

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество  
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Кафедра Информационные системы

«Допущен к защите»  
Заведующий кафедрой \_\_\_\_\_

(Ф.И.О., ученая степень, звание)

« \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ г.  
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Разработка информационной системы  
« Система управления конкуренцией »

Специальность Информатика 58060200

Выполнил (а) Юсуп Зинна Байсеитов Группа - 10-1  
(Фамилия и инициалы) группа

Научный руководитель Алмасулатова К.Б., магистр пер. курс, ст. преп.  
(Фамилия и инициалы, ученая степень, звание)

Консультанты:

по экономической части:

Бекжанова А.М., к.т.н., доцент  
(Фамилия и инициалы, ученая степень, звание)  
« 29 » мая 2014 г.  
(подпись)

по безопасности жизнедеятельности:

Диримов Н.С., д.т.н., профессор  
(Фамилия и инициалы, ученая степень, звание)  
« 24 » 05 2014 г.  
(подпись)

по применению вычислительной техники:

Жаңыбаева Б.К., м.т.н., ст. преподаватель  
(Фамилия и инициалы, ученая степень, звание)  
« 03 » 06 2014 г.  
(подпись)

Нормоконтролер: Жу А.Т., к.т.н., доцент

(Фамилия и инициалы, ученая степень, звание)  
« 09 » 06 2014 г.  
(подпись)

Рецензент:

(Фамилия и инициалы, ученая степень, звание)  
« \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ г.  
(подпись)

Алматы 2014 г.

Некоммерческое акционерное общество  
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Факультет Информационные технологии  
Специальность Информатика  
Кафедра Информационные системы

ЗАДАНИЕ

на выполнение дипломного проекта

Студент Аюсен Эмиль Байсенкулов  
(фамилия, имя, отчество)

Тема проекта Разработка информационной системы  
«Система управления конкуренцией»

утверждена приказом ректора № 115 от «14» сентября 20 13 г.

Срок сдачи законченной работы «  »    20    г.

Исходные данные к проекту требуемые параметры результатов проектирования (исследования) и исходные данные объекта

Исходные данные: задания к дипломному проекту.  
Требуемые параметры результатов проектирования:  
модель интеллектуальной информационной системы  
поддержки принятия решений, ER-диаграмма базы  
данных.

Исходные данные объекта: результаты обследования  
предметной области.

Перечень подлежащих разработке дипломного проекта вопросов или краткое содержание дипломного проекта:

1. Архитектура системы конкуренции
2. Разработка системы конкуренции
3. Внедрение системы
4. Экономическая часть
5. Безопасность жизнедеятельности

Перечень графического материала (с точным указанием обязательных чертежей)

Рисунок 1.1 - Пример диаграммы прецедентов

Рисунок 2.1 - Панель администратора, которая содержит эту функцию

Рисунок 3.1 - Пример шаблона MVC

Рекомендуемая основная литература

1. Тамми Э., Хейли Р., Джонсон Р., Виллиамс Дж. Приемы объектного проектирования. Паттерны проектирования. СПб.: Питер, 2010г.

Консультанты по работе с указанием относящихся к ним разделов

| Раздел        | Консультант      | Сроки            | Подпись     |
|---------------|------------------|------------------|-------------|
| Эконом. часть | Бекмурза А.У     |                  | <i>А.У.</i> |
| БЖД           | Турраббеков И.Г. | 11.04 - 27.05.14 | <i>И.Г.</i> |
|               |                  |                  |             |
|               |                  |                  |             |
|               |                  |                  |             |
|               |                  |                  |             |

**Г Р А Ф И К**  
подготовки дипломной работы

| № п/п | Наименование разделов, перечень разрабатываемых вопросов   | Сроки представления руководителю | Примечание |
|-------|--|----------------------------------|------------|
| 1.    | Архитектура системы конференции. Описание предметной области, структура базы данных MySQL, диаграммы прецедентов | 10.07.14-10.08.14                |            |
| 2.    | Разработка системы конференции. Разработка итерации сборщика, назначение ресурсов                                | 12.02.14-12.04.14                |            |
| 3.    | Разработка серверной части. Использование технологий. Разработка на Web-сервере Apache Tomcat                    | 14.04.14-25.05.14                |            |
| 4.    | Функциональная часть. Программное обеспечение, используемое на работе. Расчет алгоритмов отчисления              | 27.05.14-29.05.14                |            |
| 5.    | Безопасность. Проверка целостности. Таблица микромашинного и расчет безопасности и уровня шума                   | 04.03.14-4.04.14                 |            |
|       |  |                                  |            |
|       |  |                                  |            |
|       |  |                                  |            |
|       |  |                                  |            |
|       |  |                                  |            |
|       |  |                                  |            |
|       |  |                                  |            |

Дата выдачи задания « \_\_\_ » \_\_\_\_\_ 20\_\_ г.

Заведующий кафедрой \_\_\_\_\_  
(подпись) (Фамилия и инициалы)

Руководитель \_\_\_\_\_  
(подпись) (Фамилия и инициалы)

Задание принял к исполнению студент \_\_\_\_\_  
(подпись) (Фамилия и инициалы)

**Андатпа**

Конференцияны басқару жүйесі, мақалалар жинағын әзірлеу, автоматты түрде сын берушіні тағайындау, баяндаманы жіберу, конференцияны бақылау.

Берілген дипломдық жобаның зерттеу нысаны – конференцияны басқару. Жобаның мақсаты – конференцияны басқару жүйесін әзірлеу, ал бұл өз кезегінде университеттерге үйлесімді. Бұл жүйе конференцияны басқару процессін жеңілдетіп, осы процесті жақсартады. Дипломдық жұмыстың нәтижесі – конференцияны басқаруға арналған веб-қосымша. Жүйе Java программалық тіліндегі MVC паттернді Stripes фреймворкының көмегімен әзірленді. Ал жүйенің дерекқоры және мәліметтерді сақтау нысаны болып MySQL қолданылады. Веб-қосымша Apache Tomcat 6 веб-серверінде іске қосылды.

### **Аннотация**

Система управления конференцией, генерация сборника, автоматическое назначение рецензента, подача документов, мониторинг конференций.

Объектом исследования данного дипломного проекта является управление конференцией. Цель проекта – разработка системы управления конференцией, которая в частности подходит для университетов. Данная система позволит улучшить процесс управления конференцией.

Результат дипломного проекта – веб-приложение для управления конференцией. Данная система разработана с помощью MVC паттерн фреймворка Stripes, и написан на языке программирования – Java. Также использует MySQL как базу данных и объектом для хранения данных. Веб-приложение запущена на веб-сервере Apache Tomcat 6.

### **Annotation**

Conference management system, generating proceedings, automatic review assignment, paper submission, monitoring conferences.

The object of the research is a conference management. The purpose of this diploma work is to develop conference management system which is particularly suited for universities. This system will improve the process of conferences management. The result of this project is web-based conference management system. The system is developed within Stripes MVC pattern framework in Java programming language, and uses MySQL as database and object for data storage. The web-site runs in Apache Tomcat 6 web-server.

### **Содержание**

Введение

7

|      |  |    |
|------|--|----|
| 1    | Архитектура системы конференции                      | 10 |
| 1.1  | Диаграмма прецедентов                                | 10 |
| 1.2  | Структура базы данных MySQL                          | 12 |
| 1.3  | Entity Relationship Model                            | 14 |
| 2    | Разработка системы конференции                       | 21 |
| 2.1  | Генерация сборника                                   | 21 |
| 2.2  | Назначение рецензента                                | 22 |
| 2.3  | Модуль Учитель                                       | 23 |
| 2.4  | Умный Поиск  | 24 |
| 3    | Использованные технологии                            | 27 |
| 3.1  | Язык программирования Java                           | 27 |
| 3.2  | MVC  | 28 |
| 3.3  | Stripes  | 31 |
| 3.4  | Javascript   | 32 |
| 3.5  | jQuery   | 33 |
| 3.6  | iText  | 34 |
| 3.7  | CSS  | 35 |
| 3.8  | Hibernate  | 36 |
| 3.9  | Веб-сервер Apache Tomcat                             | 38 |
| 4    | Экономическая часть                                  | 40 |
| 4.1  | Трудовые ресурсы, используемые в работе              | 40 |
| 4.2  | Оборудование, используемое на работе                 | 40 |
| 4.3  | Программное обеспечение, используемое на работе      | 41 |
| 4.4  | Сроки реализации проекта                             | 41 |
| 4.5  | Затраты на разработку системы                        | 42 |
| 4.6  | Расчет фонда оплаты труда                            | 42 |
| 4.7  | Расчет затрат по социальному налогу                  | 44 |
| 4.8  | Расчет амортизационных отчислений                    | 45 |
| 4.9  | Расчет затрат на электроэнергию и накладных расходов | 45 |
| 4.10 | Цена реализации                                      | 48 |
| 5    | Безопасность жизнедеятельность                       | 50 |
| 5.1  | Характеристика условий труда программиста            | 50 |
| 5.2  | Окраска и коэффициенты отражения                     | 51 |
| 5.3  | Параметры микроклимата                               | 53 |
| 5.4  | Шум и вибрация                                       | 54 |
| 5.5  | Электромагнитное и ионизирующее излучения            | 55 |
| 5.6  | Расчет освещенности и уровня шума                    | 61 |
|      | Заключение   | 65 |
|      | Список использованной литературы                     | 66 |
|      | Приложение А   | 67 |

## **Введение**

В настоящее время каждый год проводятся многие конференции. Но подавляющее большинство из них проводятся в бумажной форме, это означает, что они не используют автоматизированную систему для регистрации, обработки запросов, статьи, процессов рассмотрения и формирования сборника. Например, все конференции во всех университетах нашей страны. Текущий процесс этих конференций полностью основан на бумажной форме. Участники подаются на этот конкурс, посылая электронные письма. Более того, они отправляют свои работы этим путем. Затем такие работы обрабатываются вручную в Excel или другие формы документа. Также администратор вручную собирает все работы и доклады участников, и ставит все это в один документ, и генерирует сборник конференции. Ежедневный сбор и проверка работы авторов, а также переработки и генерация докладов занимает более 3 дней. В связи с высокой вероятностью ошибки во время ручной обработки данных из более чем 50 форм и рецензирования материалов, ответственные за конференцию должны проверить несколько раз окончательную информацию. Много информации должны быть введены вручную или загружены из разных источников, таких как Microsoft Doc files or PDF files.

Таким образом, этот дипломный проект сосредоточен на реализации автоматизированной системы управления конференция, которая подходит для конференций в учебных заведениях. Эта система улучшит процесс в управлении конференций, конкурсов и значительно снизить нагрузку на администраторов конференции.

Более того, эта система будет решать другую типичную проблему в любом университете. В частности, многие студенты сталкиваются с проблемой в поиске руководителя для своего проекта или руководителя для дипломной работы. Потому что вы не знаете, в какой области определенные учителя заинтересованы, и в какой сфере они делают исследования. Так, учитель может загружать свои статьи, работы, научные публикации в нашу систему. Таким образом, вы можете найти руководителя, с которым у вас пересекаются интересы.

Новизна этого дипломного проекта состоит в следующем: интегрированная система, которая обеспечивает преимущества для всех заинтересованных сторон: студентов, преподавателей и владельцев конференции. Система хранит доклады, статьи и научные исследования в одном месте.

Система управления конференциями это веб-приложение, которое поддерживает организацию конференций, в особенности научные конференции. Но система не будет только для проведения конференций, но и для проведения различных онлайн соревнований и хранения статьи преподавателей и оказание помощи в выборе руководителя проектов и дипломных работ.

Сейчас существуют несколько систем управления конференциями, такие как EasyChair и Microsoft Conference Management System. Они оба

автоматизированные онлайн решения для управления и проведения конференций и семинаров. Эти системы являются бесплатными, а также может обрабатывать сложные научные конференции. Эти услуги по управлению конференциями обеспечивают большую поддержку председателям программы для управления конференц-процессами, включая настройки свойств конференции, таких как сроки, работы автора и форм рецензента. Но, несмотря на такие преимущества, они имеют ряд недостатков. Например, поддержка системы Microsoft Conference Management System ограничено. Также, ваш запрос для пользования этой услугой для управления конференций обрабатывается в течение 2-3 рабочих дней. Кроме того, они не поддерживают функцию для генерации сборника.

Анализируя существующие услуги по управлению конференциями, я нашел преимущества и недостатки каждой системы. Таким образом, мы решили включить функцию для генерации сборника к нашей системе управления конференциями.

Система состоит из нескольких функции, а именно:

а) управление конференциями - это функция позволяет создавать и управлять конференциями и семинарами. Администратор расписывает важные даты. Участники регистрируются на конференцию и представляют свои работы;

б) управление онлайн соревнованиями - это функция с помощью которой легко создавать разные интернет-проект соревнования;

в) модуль учителя - базы данных учителей, где они могут представлять статьи и научные работы. Это поможет студентам определить исследовательские интересы учителей и найти лучшего руководителя для себя.

Дипломный проект состоит из пяти глав. В первой главе описаны основные назначения диаграмм – описание функциональности и поведения, позволяющее заказчику, конечному пользователю и разработчику совместно проектировать существующую систему. Вторая часть посвящена разработки системы конференции, а именно генерации сборника, назначение рецензента, разработке модуля учителя. В третьей части описаны использованные мною технологии программирования, как Java фреймворка Stripes, который в свою очередь использует шаблон проектирования MVC (Model View Controller). В четвертой части я проанализировала экономическую реализацию проекта, отражающая временные, трудовые и финансовые затраты на проект. Пятая часть посвящена рассмотрению оптимальных условий труда инженера, расчету освещенности и расчету уровня шума.

Ожидаемые результаты работы:

- повышение качества, эффективности, результативности;
- уменьшить нагрузку администраторов конференции;
- генерация сборника конференций;
- легкое управление конференциями;
- планирование программы конференции;



- регистрация, администрирование и приглашение участников;
- процесс представления и рассмотрения взносов;
- собирает все статьи и работы преподавателей.

## **1 Архитектура системы конференции**

### **1.1 Диаграмма прецедентов**

Диаграмма прецедентов в UML — диаграмма, отражающая отношения между клиентом и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне. Прецедент — возможность моделируемой системы (часть её функциональности), благодаря которой пользователь может получить конкретный, измеримый и нужный ему результат. Прецедент соответствует отдельному сервису системы, определяет один из вариантов её использования и описывает типичный способ взаимодействия пользователя с системой. Варианты использования обычно применяются для спецификации внешних требований к системе [1].

Основное назначение диаграммы — описание функциональности и поведения, позволяющее заказчику, конечному пользователю и разработчику совместно обсуждать проектируемую или существующую систему.

Из Рисунка 1.1 видно, что клиент имеет прямую связь и имеет отношение к прецеденту заказ продукта.



Рисунок 1.1 – Пример диаграммы прецедентов

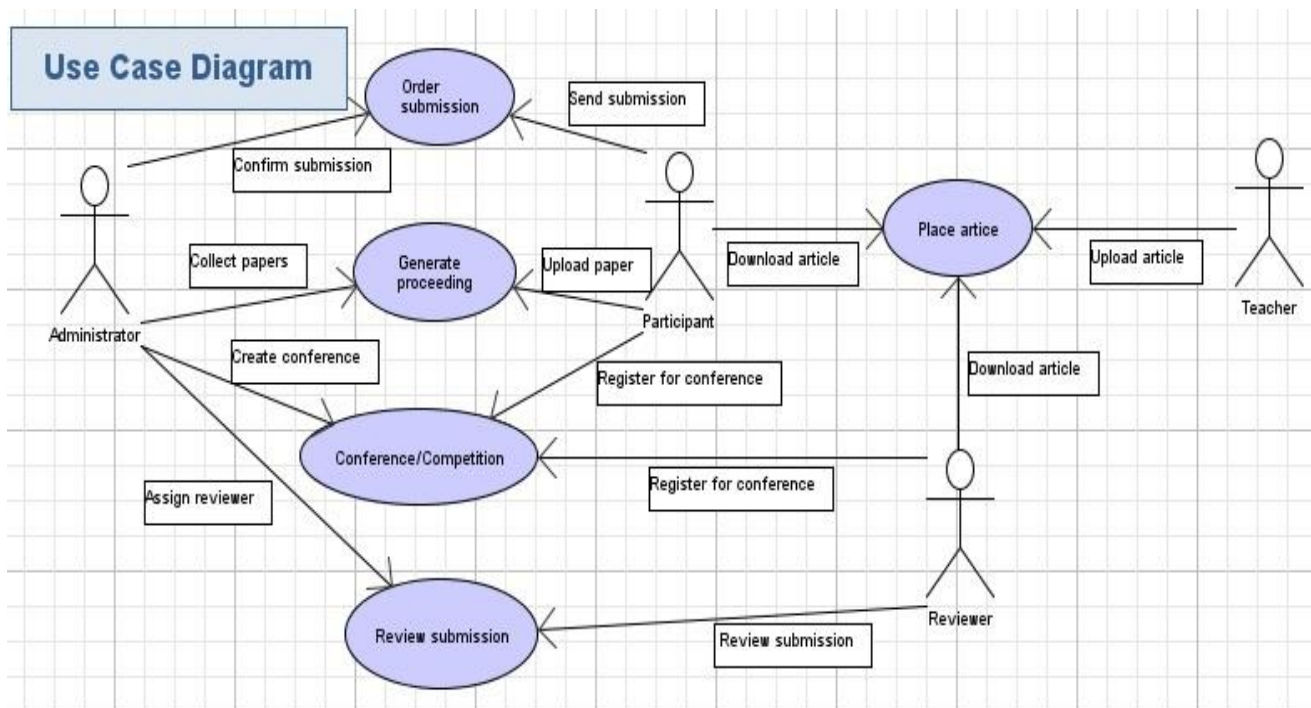


Рисунок 1.2 – Структура модели

В нашем проекте мы имеем закупщиков. Это: администратор конференции/соревнования (administrator), участник конференции/соревнования (participant), учитель (teacher), и рецензент работ, докладов, и статей (reviewer). В системе 5 прецедентов:

- подача работы и статьи (order submission);
- генерация сборника (generate proceeding);
- создать конференцию/соревнование (conference/competition);
- рецензия статьи (review submission);
- загрузка работы и научных исследований (place article).

Теперь рассмотрим подробнее каждый прецедент и дадим полное описание:

а) создать конференцию/соревнование в системе администратор конференции можно создать новую конференцию, семинар или соревнование. После регистрации конференций в системе он управляет эту конференцию, добавляет дополнительную информацию, обновляет данные или удаляет существующую информацию. К тому же, он видит всех участников конференций и делает рассылку новостей всем участникам;

б) одной стороны участник тоже играет роль в этом прецеденте. Как только администратор создаст конференцию в системе, он может зарегистрироваться в эту конференцию и подать заявку на участие;

в) после регистрации в конференции и семинаров участники подают свои работы и статьи. У каждой конференции свои правила для работ и докладов. После администратор и модератор мероприятия проверяют загруженные работы по своим правилам, и подтверждают подачу, либо отклоняют;

г) рецензия статьи в конференциях, где участвуют тысячи участников и работы, комитету трудно оценивать работу вручную. Исследуется применение методов машинного обучения для решения этой задачи, а именно, учитывая информацию некоторых конференциях и свои задания, мы будем выполнять эффективную оценку параметров и использовать этот модель для вывода задания в ряде других конференций. Учитывая конференцию с участием М рецензентов и N представленных документов, мы хотели бы выделить каждому рецензенту один доклад для оценки, такой, чтоб каждый рецензент адекватно давал отзывы. В этом прецеденте мы имеем двух актеров – рецензента и администратора;

д) генерация сборника является одним из важных функции системы. Ручная генерация сборников требует много времени. Ты должен собрать все работы участников и копировать их всех в один файл. Это трудная задача для модератора. Итак, с помощью этой функций мы уменьшили нагрузку и сделали генерацию сборника легкой. Эта функция была разработана с помощью библиотеки iText, которая используется для работы с PDF файлами.

ж) загрузка работы и научных исследований это система управления конференциями хранит информацию об учителей. У них есть своя отдельная страница, где они могут управлять ею. Также, они могут загружать документы, статьи, и свои работы.

## 1.2 Структура базы данных MySQL

MySQL - свободная реляционная система управления базами данных. Корпорация Oracle осуществляет разработку и поддержку MySQL, которая получила права на торговую марку вместе с поглощённой Sun Microsystems. Ранее Sun Microsystems приобрела шведскую компанию MySQL AB. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией.

Существует несколько типов моделей баз данных. Иерархические и сетевые крайне не удобны в работе. Тогда как реляционные получили широкое признание. Реляционная база данных впервые появилась на свет в 1969 году и бесспорно стала сегодня наиболее широко используемой моделью базы данных в системах управления базами данных. Эдгар Ф. Кодд официально представил свою новую реляционную модель в эпохальной работе, озаглавленной «A Relational Model of Data for Large Shared Databanks - реляционная модель данных для больших совместно используемых банков данных». В основу своей новой модели он положил два раздела математики: теорию множеств и исчисление предикатов первого порядка. Само наименование новой модели происходит от термина «отношение», заимствованного из теории множеств. С момента появления реляционная модель является основой программных продуктов для баз данных, известных как система управления реляционными базами данных. Выпускаемые

различными поставщиками, они спустя несколько лет получили признание в различных отраслях и организациях и используются в средах различного типа.

В 70-х годах мэйнфреймы использовали такие программы, как System R компании IBM и INGRES, разработанную в Калифорнийском университете в Беркли. В 80-8 годы СУРБД для мэйнфреймов получили развитие таких программных продуктах, как Oracle одноименной компании и DB2 компании IBM. Бурное развитие персональных компьютеров в середине 80-х вызвало появление таких программ, как dBase от Ashton Tate, Paradox от Ansa Software и R:BASE от Microrim. Когда в начале 90-х годов стала очевидной потребностью в совместном использовании данных персональными компьютерами, родилась клиент-серверная вычислительная модель вместе с идеей централизованно расположенных общих данных, которыми было бы легко управлять и обеспечивать из безопасность. Эта концепция дала начало таким продуктам, как Oracle 8i и Microsoft SQL Server 7.

Приблизительно с 1996 года были приложены более согласованные усилия к обеспечению доступности баз данных в Интернете. Поставщики программного обеспечения серьезно отнеслись к этим усилиям и теперь предлагают продукты «с расширенной поддержкой Web», такие как Cold Fusion от Allaire, Sybase Enterprise Application Studio от Sybase и Visual InterDev от Microsoft.

В соответствии с реляционной моделью данные в реляционной базе данных сохраняются в отношениях, или связях, которые воспринимаются пользователем как таблицы. Каждое отношение состоит из кортежей (записей) и атрибутов (полей).

Таблицы являются основными структурами в базе данных. Каждая таблица описывает отдельный предмет. Логический порядок записей и полей в таблице совершенно не имеет значения. Каждая таблица содержит хотя бы одно поле (называемое первичным ключом), которое однозначно идентифицирует каждую из записей.

Поле является атомарной структурой в базе данных, и оно представляет собой характеристику предмета таблицы, к которой оно относится. Поля реально используются для хранения данных. Данные из этих полей можно затем извлекать и представлять как информацию почти в любой вообразимой форме. Качество информации получаемой из таблицы прямо пропорционально времени посвященному обеспечению структурной целостности и целостности данных в самих полях. Совершенно невозможно недооценить важность полей.

Каждое поле в базе данных, спроектированной надлежащим образом, содержит одно и только одно значение, а имя поле идентифицирует тип хранимого в нем значения. Это само по себе делает ввод данных в поле интуитивным. Если поля имеют такие имена, как FirstName, LastName, City, State и ZipCode, то не нужно гадать, значения какого типа заносятся в каждое поле. Также облегчается процесс сортировки данных по состоянию или поиск какого-либо с фамилией Hernandez.

Запись представляет собой уникальный экземпляр предмета таблицы. Она составляется из полного набора полей в таблице независимо от того, содержат ли эти поля какие-либо значения или нет. Вследствие способа определения таблицы каждая запись идентифицируется в базе данных уникальным значением в поле первичного ключа этой записи.

### 1.3 Entity Relationship Model

Entity Relationship Model (русс. Модель сущность-связь, ERM) — модель данных, позволяющая описывать концептуальные схемы предметной области.

ER-модель используется при высокоуровневом (концептуальном) проектировании баз данных. С её помощью можно выделить ключевые сущности и обозначить связи, которые могут устанавливаться между этими сущностями.

Во время проектирования баз данных происходит преобразование ER-модели в конкретную схему базы данных на основе выбранной модели данных (реляционной, объектной, сетевой и др.).

ER-модель представляет собой формальную конструкцию, которая сама по себе не предписывает никаких графических средств её визуализации. В качестве стандартной графической нотации, с помощью которой можно визуализировать ER-модель, была предложена диаграмма сущность-связь (ER-диаграмма) (англ. entity-relationship diagram, ERD).

Модель «сущность-связь» была предложена в 1976 году Питером Пин-Шен Ченом (англ. Peter Pin-Shen Chen), американским профессором компьютерных наук в университете штата Луизиана [3].

Множества сущностей изображаются в виде прямоугольников, множества отношений изображаются в виде ромбов. Если сущность участвует в отношении, они связаны линией. Если отношение не является обязательным, то линия пунктирная. Атрибуты изображаются в виде овалов и связываются линией с одним отношением или с одной сущностью [3].

Данная нотация была предложена Гордоном Эверестом (англ. Gordon Everest) под названием Inverted Arrow («перевёрнутая стрелка»), однако сейчас чаще называемая Crow's Foot («воронья лапка») или Fork («вилка») [4].

Согласно данной нотации, сущность изображается в виде прямоугольника, содержащем её имя, выражаемое существительным. Имя сущности должно быть уникальным в рамках одной модели. При этом, имя сущности — это имя типа, а не конкретного экземпляра данного типа. Экземпляр сущности называется конкретный представитель данной сущности.

Связь изображается линией, которая связывает две сущности, участвующие в отношении. Степень конца связи указывается графически, множественность связи изображается в виде «вилки» на конце связи. Модальность связи так же изображается графически — необязательность

связи помечается кружком на конце связи. Именование обычно выражается одним глаголом в изъявительном наклонении настоящего времени: «Имеет», «Принадлежит» и т. д.; или глаголом с поясняющими словами: «Включает в себя». Наименование может быть одно для всей связи или два для каждого из концов связи. Во втором случае, название левого конца связи указывается над линией связи, а правого – под линией. Каждое из названий располагается рядом с сущностью, к которой оно относится.

Атрибуты сущности записываются внутри прямоугольника, изображающего сущность и выражаются существительным в единственном числе (возможно, с уточняющими словами). Среди атрибутов выделяется ключ сущности — не избыточный набор атрибутов, значения которых в совокупности являются уникальными для каждого экземпляра сущности.

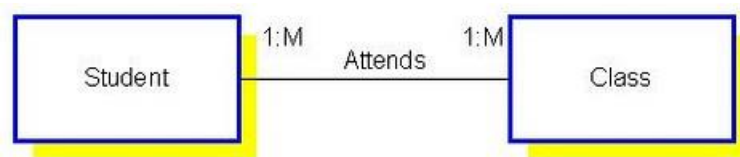


Рисунок 1.3 – Пример простой E/R Модели

Теперь рассмотрим entity relationship модель системы управления конференциями. В базе данных у нас есть 5 таблиц.



Рисунок 1.4 – Entity relationship модель таблиц

Теперь рассмотрим рисунок 1.5 E/R Модель между таблицами и полями.

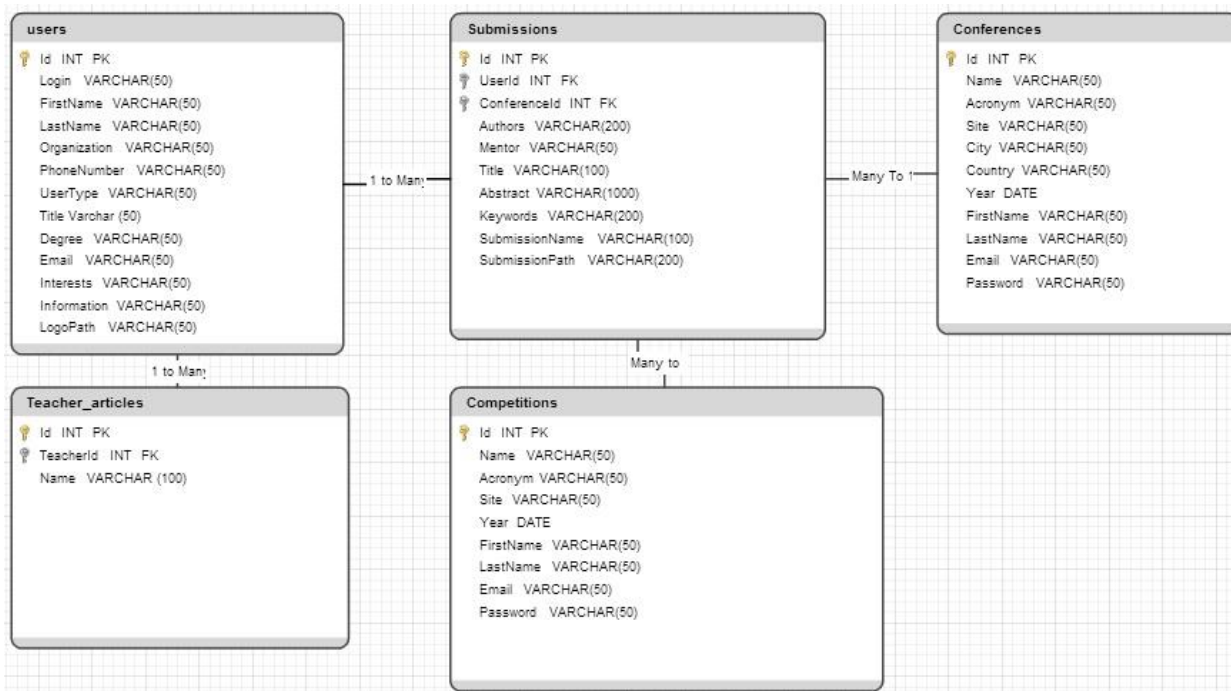


Рисунок 1.5 – E/R Модель между таблицами.

База данных - это организованная коллекция данных [2]. База данных для системы управления конференциями называется “conference\_system”, который InnoDB тип и с сравнением “utf8\_unicode\_ci”.

| Таблица          | Действие | Записи | Тип    | Сравнение       | Размер  | Фрагментировано |
|------------------|----------|--------|--------|-----------------|---------|-----------------|
| competitions     |          | 1      | InnoDB | utf8_unicode_ci | 16.0 KB | -               |
| conferences      |          | 2      | InnoDB | utf8_unicode_ci | 16.0 KB | -               |
| submissions      |          | 2      | InnoDB | utf8_unicode_ci | 16.0 KB | -               |
| teacher_articles |          | 4      | InnoDB | utf8_unicode_ci | 16.0 KB | -               |
| users            |          | 6      | InnoDB | utf8_unicode_ci | 16.0 KB | -               |

Рисунок 1.6 – Структура базы данных системы

Таблица – структура данных, хранящая набор однотипных записей [2]. В нашей базе мы имеем 5 таблиц, и у всех сравнение “utf8\_unicode\_ci”. Таблица “users” хранит информацию о пользователей системы. Новая запись добавляется, когда пользователь регистрируется в системе. В системе два вида пользователей – учителя и студенты. Система определяет пользователя константным значением.



| Поле                                  | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  | Действие |
|---------------------------------------|--------------|-----------------|----------|------|--------------|----------------|----------|
| <input type="checkbox"/> id           | int(11)      |                 |          | Нет  | Нет          | AUTO_INCREMENT |          |
| <input type="checkbox"/> login        | varchar(50)  | utf8_unicode_ci |          | Нет  | Нет          |                |          |
| <input type="checkbox"/> firstName    | varchar(50)  | utf8_unicode_ci |          | Нет  | Нет          |                |          |
| <input type="checkbox"/> lastName     | varchar(50)  | utf8_unicode_ci |          | Нет  | Нет          |                |          |
| <input type="checkbox"/> organization | varchar(50)  | utf8_unicode_ci |          | Нет  | Нет          |                |          |
| <input type="checkbox"/> phoneNumber  | varchar(50)  | utf8_unicode_ci |          | Нет  | Нет          |                |          |
| <input type="checkbox"/> userType     | varchar(50)  | utf8_unicode_ci |          | Нет  | Нет          |                |          |
| <input type="checkbox"/> title        | varchar(50)  | utf8_unicode_ci |          | Да   | NULL         |                |          |
| <input type="checkbox"/> degree       | varchar(50)  | utf8_unicode_ci |          | Да   | NULL         |                |          |
| <input type="checkbox"/> password     | varchar(50)  | utf8_unicode_ci |          | Нет  | Нет          |                |          |
| <input type="checkbox"/> email        | varchar(50)  | utf8_unicode_ci |          | Нет  | Нет          |                |          |
| <input type="checkbox"/> interests    | text         | utf8_unicode_ci |          | Нет  | Нет          |                |          |
| <input type="checkbox"/> information  | text         | utf8_unicode_ci |          | Нет  | Нет          |                |          |
| <input type="checkbox"/> logoPath     | varchar(200) | utf8_unicode_ci |          | Нет  | Нет          |                |          |

Рисунок 1.7 – Структура таблицы “users”

Поля таблицы:

- id – уникальный номер каждого пользователя. Это первичный ключ таблицы;
- login – уникальное имя пользователя каждого пользователя. Имя пользователя содержит имя и фамилию пользователя;
- first name – имя пользователя;
- last name – фамилия пользователя;
- organization – название организаций;
- phone number – номер телефона пользователя;
- user type – поля, которое определяет тип пользователя.
- title – титул пользователя, только для учителей;
- degree – степень пользователя, только для учителей;
- email – почтовый адрес пользователя;
- password – пароль, используется для аутентификации пользователя;
- interests – интересы пользователя;
- information – содержит дополнительную информацию;
- logo path – путь до лого пользователя.

Таблица “conferences” содержит информацию про конференции и семинары. Новая запись добавляется, когда регистрируется новая конференция.

| Поле                               | Тип         | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  | Действие |
|------------------------------------|-------------|-----------------|----------|------|--------------|----------------|----------|
| <input type="checkbox"/> id        | int(11)     |                 |          | Нет  | Нет          | AUTO_INCREMENT | [Иконки] |
| <input type="checkbox"/> name      | varchar(50) | utf8_unicode_ci |          | Нет  | Нет          |                | [Иконки] |
| <input type="checkbox"/> acronym   | varchar(50) | utf8_unicode_ci |          | Нет  | Нет          |                | [Иконки] |
| <input type="checkbox"/> site      | varchar(50) | utf8_unicode_ci |          | Да   | NULL         |                | [Иконки] |
| <input type="checkbox"/> city      | varchar(50) | utf8_unicode_ci |          | Нет  | Нет          |                | [Иконки] |
| <input type="checkbox"/> country   | varchar(50) | utf8_unicode_ci |          | Нет  | Нет          |                | [Иконки] |
| <input type="checkbox"/> year      | int(11)     |                 |          | Нет  | Нет          |                | [Иконки] |
| <input type="checkbox"/> firstName | varchar(50) | utf8_unicode_ci |          | Нет  | Нет          |                | [Иконки] |
| <input type="checkbox"/> lastName  | varchar(50) | utf8_unicode_ci |          | Нет  | Нет          |                | [Иконки] |
| <input type="checkbox"/> email     | varchar(50) | utf8_unicode_ci |          | Нет  | Нет          |                | [Иконки] |
| <input type="checkbox"/> password  | varchar(50) | utf8_unicode_ci |          | Нет  | Нет          |                | [Иконки] |

Рисунок 1.8 – Структура таблицы “conferences”

Поля таблицы:

- id – уникальный номер каждой конференций. Это первичный ключ таблицы;

- name – название конференции;

- acronym – уникальное название конференции;

- site – веб-сайт конференции;

- city – город, в котором проводится конференция;

- country – страна, в котором проводится конференция;

- year – дата проведения;

- first name – имя администратора;

- last name – фамилия администратора;

-email – адрес почтового ящика;

- password – пароль для аутентификации.

Таблица “competitions” содержит информацию про состязания. Новая запись добавляется когда регистрируется новое состязание.

| Поле                               | Тип         | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  | Действие |
|------------------------------------|-------------|-----------------|----------|------|--------------|----------------|----------|
| <input type="checkbox"/> id        | int(11)     |                 |          | Нет  | Нет          | AUTO_INCREMENT | [Иконки] |
| <input type="checkbox"/> name      | varchar(50) | utf8_unicode_ci |          | Нет  | Нет          |                | [Иконки] |
| <input type="checkbox"/> acronym   | varchar(50) | utf8_unicode_ci |          | Нет  | Нет          |                | [Иконки] |
| <input type="checkbox"/> site      | varchar(50) | utf8_unicode_ci |          | Нет  | Нет          |                | [Иконки] |
| <input type="checkbox"/> year      | int(11)     |                 |          | Нет  | Нет          |                | [Иконки] |
| <input type="checkbox"/> firstName | varchar(50) | utf8_unicode_ci |          | Нет  | Нет          |                | [Иконки] |
| <input type="checkbox"/> lastName  | varchar(50) | utf8_unicode_ci |          | Нет  | Нет          |                | [Иконки] |
| <input type="checkbox"/> email     | varchar(50) | utf8_unicode_ci |          | Нет  | Нет          |                | [Иконки] |
| <input type="checkbox"/> password  | varchar(50) | utf8_unicode_ci |          | Нет  | Нет          |                | [Иконки] |

Рисунок 1.9 – Структура таблицы “competitions”

Поля таблицы:

- id – уникальный номер каждого соревнования. Это первичный ключ таблицы;

- name – полное название состязания;
- acronym – уникальное название состязания;
- site – веб-сайт;
- year – дата проведения;
- first name – имя администратора;
- last name – фамилия администратора;
- email – адрес почтового ящика;
- password – пароль для аутентификации.

Таблица “submissions” хранит работы участников конференции. Все работы и доклады хранятся в этой таблице.

| Поле                                    | Тип           | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  | Действие |
|---|---------------|-----------------|----------|------|--------------|----------------|----------|
| <input type="checkbox"/> id             | int(11)       |                 |          | Нет  | Нет          | AUTO_INCREMENT | [Иконки] |
| <input type="checkbox"/> userId         | int(11)       |                 |          | Нет  | Нет          |                | [Иконки] |
| <input type="checkbox"/> authors        | varchar(200)  | utf8_unicode_ci |          | Нет  | Нет          |                | [Иконки] |
| <input type="checkbox"/> mentor         | varchar(50)   | utf8_unicode_ci |          | Нет  | Нет          |                | [Иконки] |
| <input type="checkbox"/> title          | varchar(100)  | utf8_unicode_ci |          | Нет  | Нет          |                | [Иконки] |
| <input type="checkbox"/> abstract       | varchar(1000) | utf8_unicode_ci |          | Нет  | Нет          |                | [Иконки] |
| <input type="checkbox"/> keywords       | varchar(200)  | utf8_unicode_ci |          | Нет  | Нет          |                | [Иконки] |
| <input type="checkbox"/> submissionName | varchar(100)  | utf8_unicode_ci |          | Нет  | Нет          |                | [Иконки] |
| <input type="checkbox"/> submissionPath | varchar(200)  | utf8_unicode_ci |          | Нет  | Нет          |                | [Иконки] |
| <input type="checkbox"/> conferenceId   | int(11)       |                 |          | Нет  | Нет          |                | [Иконки] |

Рисунок 1.10 - Структура таблицы “submissions”

Поля таблицы:

- id – уникальный номер каждой работы. Это первичный ключ таблицы;
- user id - уникальный номер автора работы;
- authors – авторы работы;
- mentor – руководитель работы;
- title – название документа;
- abstract – аннотация работы, доклада;
- keywords – ключевые слова, которые описывают работу;
- submission name – название работы, доклада;
- submission path – путь к файлу, где хранится работа;
- conference id - уникальный номер конференции.

Таблица “teacher\_articles” содержит информация о работах, докладах, научных исследований учителей.

| Поле                               | Тип          | Сравнение       | Атрибуты | Null | По умолчанию | Дополнительно  | Действие |
|------------------------------------|--------------|-----------------|----------|------|--------------|----------------|----------|
| <input type="checkbox"/> id        | int(11)      |                 |          | Нет  | Нет          | AUTO_INCREMENT | [Иконки] |
| <input type="checkbox"/> name      | varchar(100) | utf8_unicode_ci |          | Нет  | Нет          |                | [Иконки] |
| <input type="checkbox"/> teacherId | int(11)      |                 |          | Нет  | Нет          |                | [Иконки] |

Рисунок 1.11 – Структура таблицы “teacher\_articles”

Поля таблицы:

- id – уникальный номер каждого учителя. Это первичный ключ таблицы;
- name – название документа;
- user id - уникальный номер учителя.

## 2 Разработка системы конференции

### 2.1 Генерация сборника

Функция генерация сборника позволяет администратору, модератору конференции, семинара генерировать сборника этого мероприятия. Эта функция собирает все работы и доклады определенной конференции в один документ PFD формата. Данная функция реализована с помощью библиотеки iText, которая является бесплатной и предназначена для работы и создания, управления PDF файлами на Java. Особенно, была разработана с использованием класса Pdf Copy библиотеки iText. Класс Pdf Copy копирует страницы с одного или нескольких PDF документов. Основные минусы: Pdf Copy не обнаруживает избыточное содержание, и выходит из строя при объединении форм. Мы также можем использовать классы Pdf Writer и Pdf Stamer для генерации сборника с нескольких файлов, и это не проблема. Потому что мы можем легко добавить содержимое в импортированные страницы. Но при использовании класса Pdf Copy, было бы интересно добавить дополнительный контент. Также данная функция позволяет добавлять заголовки и нижние колонтитулы. Так как это большой плюс когда мы может вставить номер страницы в нижний колонтитул для нового созданного документа.

Важная часть кода:

```
@HandlesEvent("generateWithPDFCopy")
public void generateWithPDFCopy() throws DocumentException, IOException{
    ...
    for(int i = 0; i < getSubmissionList().size(); i++) {
        files[i] = rootPdf + getSubmissionList().get(i).getSubmissionName();
    }
    ...
    for (int i = 0; i < files.length; i++) {
        PdfReader reader1 = new PdfReader(files[i]);
        totalPages = totalPages + reader1.getNumberOfPages();
    }
    for (int i = 0; i < files.length; i++) {
        reader = new PdfReader(files[i]);
        n = reader.getNumberOfPages();
        for (int page = 0; page < n;) {
            curPage = curPage + page2;
            page1 = copy.getImportedPage(reader, ++page);
            stamp = copy.createPageStamp(page1);
            ColumnText.showTextAligned(
                stamp.getUnderContent(), Element.ALIGN_CENTER,
                new Phrase(String.format("%d", page + page2)), 220.5f, 28, 0);
        }
    }
}
```

```
        stamp.alterContents();
        copy.addPage(page1);
    }
    page2 = n;
}
}
```

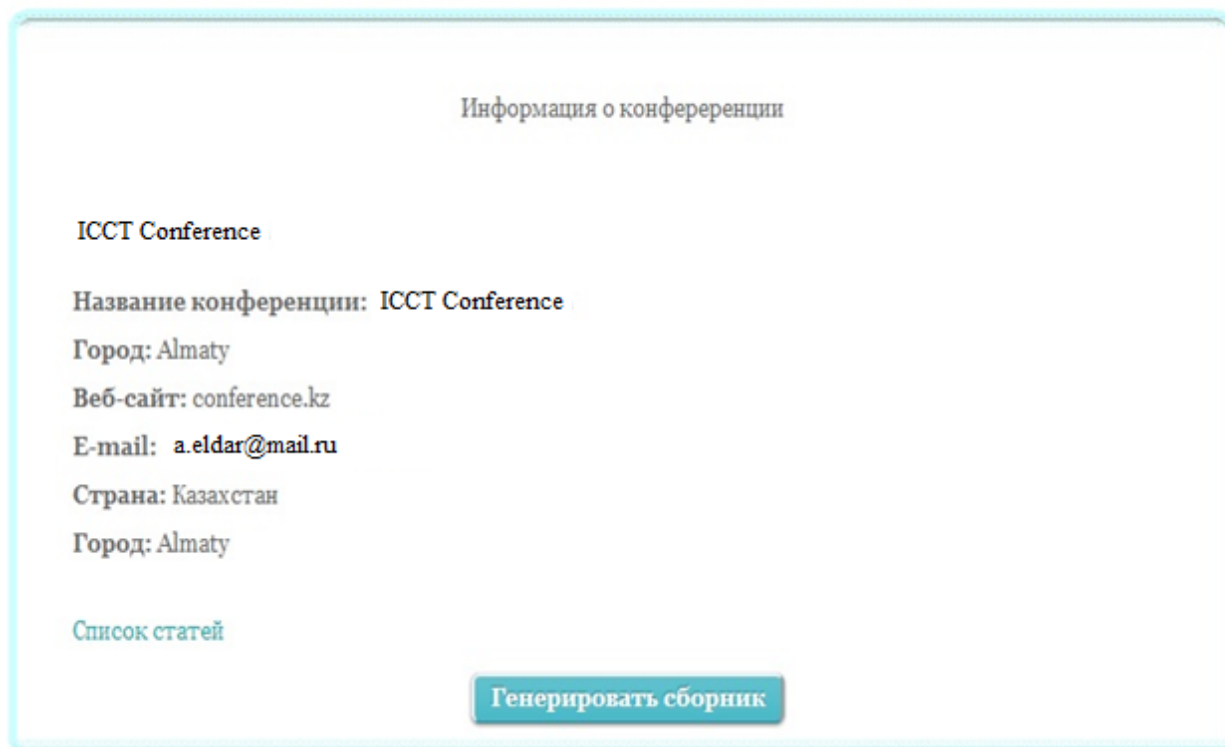


Рисунок 2.1 – Панель администратора, которая содержит эту функцию

## 2.2 Назначение рецензента

Одной из важных и стимулирующих задач при проведении научных конференции является назначение рецензентов на доклады участников. Эта аккуратность напрямую влияет на качество проведения конференции. Для высококвалифицированных упорядоченных конференций, имеющих низкий коэффициент приема, очень важно, чтобы каждый документ оценивался наиболее компетентными рецензентами и членами Комитета по программе. Даже небольшая неточность в назначении может привести к серьезным и очень неприятным последствиям, которые могут значительно уменьшить рейтинг конференции и доверие авторов в этом случае.

Назначение рецензентов на доклады вручную самый надежный вариант, по мнению многих администраторов конференции. Но это подходит для маленьких конференции, где количество участников, и соответственно количество работ маленькая.

Автоматическое назначение рецензента увеличивает аккуратность и качество научного мероприятия. Так как при возрастании количество работ и докладов ручное назначение становится все труднее и труднее. К тому же, это становится менее аккуратным и важным. Авторы работ и докладов должны описать свои работы ключевыми словами и аннотацией, чтобы улучшить аккуратность и эффективность автоматического назначения. В случаях, где имеется большое количество докладов, автоматическое назначение является более подходящим и лучшим решением.

### 2.3 Модуль Учителя

Главная цель дипломной работы разработать автоматизированную систему управления конференциями. Но в этой системе есть еще одна дополнительная функция – Модуль Учителя. Этот модуль позволяет учителям регистрироваться на сайт и загружать свои работы, статьи и другие документы. Система определяет учителей константным значением. Также они могут добавлять лого картинки, дополнительную информацию о себе – интересы, информацию о своем департаменте, степени и т.д. Они также могут изменять и обновлять свою информацию, и удалять загруженные документы. Все информация может быть полезна для студентов. На персональной странице учителя будет контактная информация, например, адрес почтового ящика или номер телефона.



Рисунок 2.2 – Список учителей

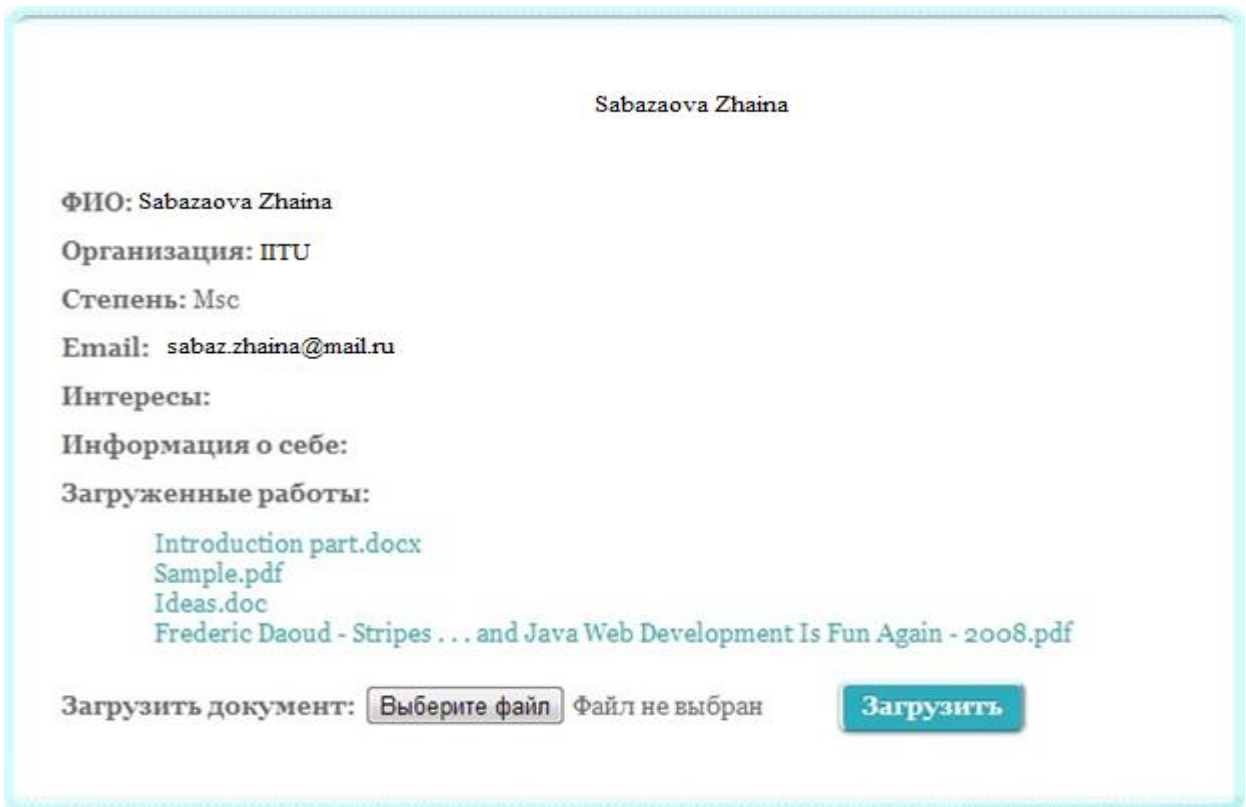


Рисунок 2.3 – Страница учителя

Важная часть кода для загрузки работы:

```
...  
TeacherArticle teacherArticle = new TeacherArticle();  
teacherArticle.setName(getArticle().getFileName());  
teacherArticle.setTeacherId(getUserId());  
String rootPdf = getContext().getServletContext()  
    .getInitParameter("TEACHERS_FILES_PATH");  
try  
{  
    article.save(new File(rootPdf + article.getFileName()));  
}  
...  
getContext().getMessages().add(new SimpleMessage("Ваша работа успешно  
    загружена"));  
return new RedirectResolution(TeacherActionBean.class, "uploadOk");
```

## 2.4 Умный Поиск

Веб-система содержит разные информации о конференциях, семинарах и онлайн-соревнованиях. В базе системы хранится очень большой объем данных. Если кто-то захочет найти информацию, которая ему нужна, это было бы большой проблемой. С помощью функции Умный Поиск мы решаем эту



проблему и легко предоставляем пользователю информацию, которая его интересует. Эта функция ищет информацию по ключевым словам в конференциях, статьях, соревнованиях и информациях учителей.



Рисунок 2.4 – Секция поиска

Для этой функции мы создали форму Java класс, которая называется “SearchForm”. У этого класса есть четыре атрибута, которые описывают его. Также мы использовали класс Criteria библиотеки Hibernate для выполнения запроса и поиска подходящей информации. Пользователь набирает ключевые слова и нажимает на кнопку поиска. Javascript функция подтверждает форму и делает запрос на контроллер-класс “SearchActionBean”. Этот класс находит данные по ключевым словам с помощью Criteria и добавляет в класс “SearchForm”.

Важная часть кода в странице JSP:

```
<stripes:form beanclass="actions.SearchActionBean" id="searchForm">
  <input type="image" src="{ctx}/images/search.png"
    onclick="submitJavaScriptForm('searchForm');" style="height:20px;"/>
</stripes:form>
```

```
<script type="text/javascript">
  function submitJavaScriptForm(formId){
    var choicePanelForm = document.getElementById(formId);
    choicePanelForm.submit();
  }
</script>
```

В классе ActionBean:

```
...
Criteria conferences = s.createCriteria(Conference.class);
conferences.add(Restrictions.or(Restrictions.ilike("name", "%" + keyword +
"%"),
Restrictions.ilike("acronym", "%" + keyword + "%")));
List<Conference> resSites = conferences.list();
for(Conference conference : resSites) {
  String siteUrl =
"http://localhost:8080/actions/Conference.action?conference_page=&conferenceId="
+ conference.getId();
```

```
        SearchForm form = new SearchForm(conference.getId(),
conference.getName(),
conference.getAcronym(), siteUrl);
        result.add(form);
    }
    ...
    return new ForwardResolution("/WEB-INF/jsp/search.jsp");
```

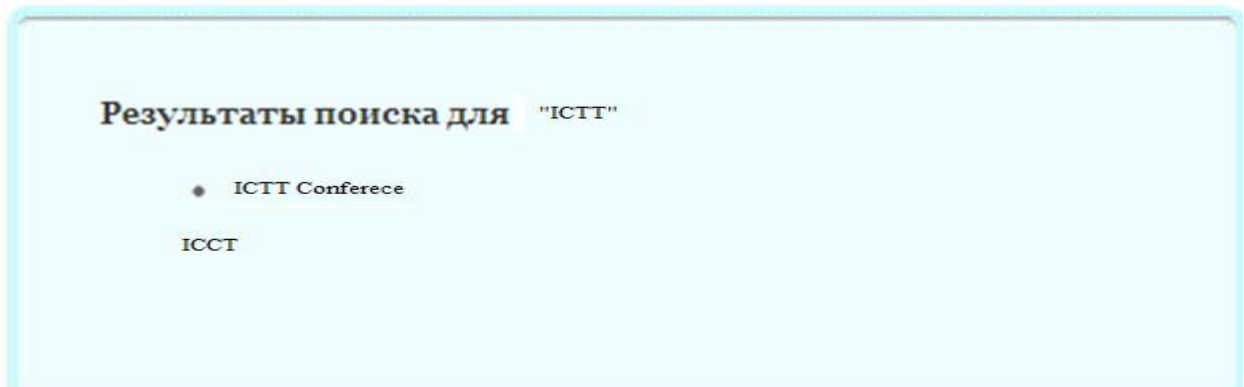


Рисунок 2.5 – Результаты поиска для ключевого слова “ICTT”

### 3 Используемые технологии

#### 3.1 Язык программирования Java

Система управления конференциями написана на языке программирования Java с помощью фреймворка Stripes, который в свою очередь использует шаблон проектирования MVC (Model-View-Controller).

Я выбрала язык программирования Java, потому что он позволяет эффективно разрабатывать, откладывать и использовать увлекательные приложения и сервисы. Также Java является основой практически для каждого типа сетевого приложения и является мировым стандартом для разработки и внедрения мобильных приложений, игр, веб-содержимого на основе и корпоративного программного обеспечения.

Java – объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle). Приложения Java обычно транслируются в специальный байт-код, поэтому они могут работать на любой виртуальной Java-машине вне зависимости от компьютерной архитектуры [2]. Программы на Java транслируются в байт-код, выполняемый виртуальной машиной Java (JVM) – программой, обрабатывающей байтовый код и передающей инструкции оборудованию как интерпретатор.

Достоинством подобного способа выполнения программ является полная независимость байт-кода от операционной системы и оборудования, что позволяет выполнять Java-приложения на любом устройстве, для которого существует соответствующая виртуальная машина. Другой важной особенностью технологии Java является гибкая система безопасности благодаря тому, что исполнение программы полностью контролируется виртуальной машиной. Любые операции, которые превышают установленные полномочия программы (например, попытка несанкционированного доступа к данным или соединения с другим компьютером) вызывают немедленное прерывание.

Основные возможности:

- автоматическое управление памятью;
- расширенные возможности обработки исключительных ситуаций;
- богатый набор средств фильтрации ввода-вывода;
- набор стандартных коллекций: массив, список, стек и т. п.;
- наличие простых средств создания сетевых приложений (в том числе с использованием протокола RMI);
- наличие классов, позволяющих выполнять HTTP-запросы и обрабатывать ответы;
- встроенные в язык средства создания многопоточных приложений;
- унифицированный доступ к базам данных;
- на уровне отдельных SQL-запросов — на основе JDBC, SQLJ;

- на уровне концепции объектов, обладающих способностью к хранению в базе данных — на основе Java Data Objects и Java Persistence API;
- поддержка обобщений (начиная с версии 1.5);
- параллельное выполнение программ.

### 3.2 MVC

Model–view–controller (MVC, «модель-представление-поведение», «модель-представление-контроллер», «модель-вид-контроллер») — схема использования нескольких шаблонов проектирования, с помощью которых модель данных приложения, пользовательский интерфейс и взаимодействие с пользователем разделены на три отдельных компонента таким образом, чтобы модификация одного из компонентов оказывала минимальное воздействие на остальные. Данная схема проектирования часто используется для построения архитектурного каркаса, когда переходят от теории к реализации в конкретной предметной области [5].

Концепция MVC была описана в 1979 году Трюгве Реенскауг (англ. Trygve Reenskaug), тогда работающим над языком программирования Smalltalk в Xerox PARC. Оригинальная реализация описана в статье «Applications Programming in Smalltalk-80: How to use Model-View-Controller». Затем Джим Алтофф с командой разработчиков реализовали версию MVC для библиотеки классов Smalltalk-80.

Концепция MVC позволяет разделить данные, представление и обработку действий пользователя на три отдельных компонента:

а) модель (англ. Model). Модель предоставляет знания: данные и методы работы с этими данными, реагирует на запросы, изменяя своё состояние. Не содержит информации, как эти знания можно визуализировать;

б) представление, вид (англ. View). Отвечает за отображение информации (визуализацию). Часто в качестве представления выступает форма (окно) с графическими элементами;

в) контроллер (англ. Controller). Обеспечивает связь между пользователем и системой: контролирует ввод данных пользователем и использует модель и представление для реализации необходимой реакции.

Важно отметить, что как «представление», так и «контроллер» зависят от «модели». Однако «модель» не зависит ни от «представления», ни от «контроллера». Тем самым достигается назначение такого разделения: оно позволяет строить «модель» независимо от «визуального представления», а также создавать несколько различных «представлений» для одной «модели».

Для реализации схемы Model-View-Controller используется достаточно большое число шаблонов проектирования (в зависимости от сложности архитектурного решения), основные из которых «наблюдатель», «стратегия», «компоновщик» [6].

Наиболее типичная реализация отделяет вид от модели путем установления между ними протокола взаимодействия, используя аппарат

событий (подписка/оповещение). При каждом изменении внутренних данных в модели она оповещает все зависящие от неё представления, и представление обновляется. Для этого используется шаблон «наблюдатель». При обработке реакции пользователя вид выбирает, в зависимости от нужной реакции, нужный контроллер, который обеспечит ту или иную связь с моделью. Для этого используется шаблон «стратегия», или вместо этого может быть модификация с использованием шаблона «команда». А для возможности однотипного обращения с подобъектами сложно-составного иерархического вида может использоваться шаблон «компоновщик». Кроме того, могут использоваться и другие шаблоны проектирования, например, «фабричный метод», который позволит задать по умолчанию тип контроллера для соответствующего вида.

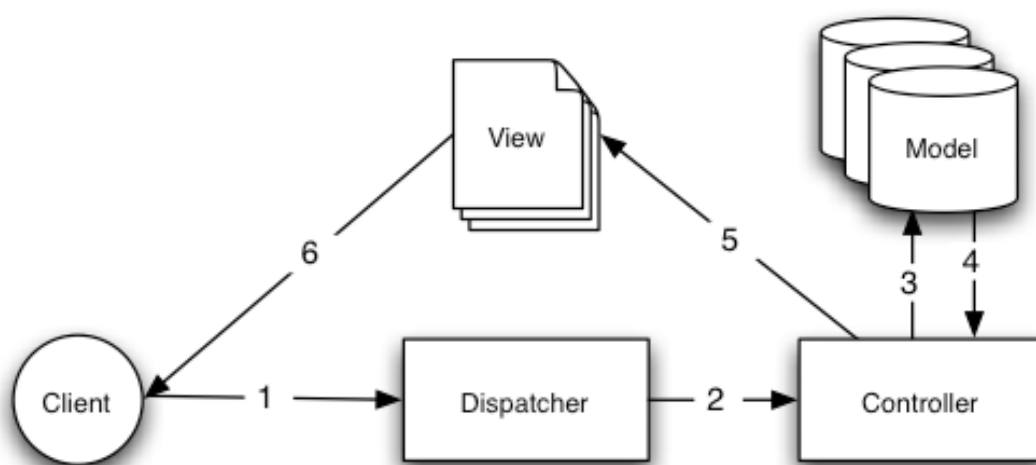


Рисунок 3.1 – Пример шаблона MVC

В этом проекте классы ActionBean - Контроллеры, JavaServer Pages (JSP) – Представления, и База Данных - Модель.

JSP (JavaServer Pages) – технология, позволяющая веб-разработчикам легко создавать содержимое, которое имеет как статические, так и динамические компоненты. По сути, страница JSP является текстовым документом, который содержит текст двух типов: статические исходные данные, которые могут быть оформлены в одном из текстовых форматов HTML, SVG, WML, или XML, и JSP элементы, которые конструируют динамическое содержимое. Кроме этого могут использоваться библиотеки JSP тегов, а также EL (Expression Language), для внедрения Java-кода в статичное содержимое JSP-страниц.

JSP – одна из высокопроизводительных технологий, так как весь код страницы транслируется в java-код сервлета с помощью компилятора JSP страниц Jasper, и затем компилируется в байт-код виртуальной машины java (JVM). Контейнеры сервлетов, способные исполнять JSP страницы, написаны на языке Java, который может работать на различных

платформах. JSP страницы загружаются на сервере и управляются из структуры специального Java server packet, который называется Java EE Web Application, в большинстве своём упакованные в файловые архивы .war и .ear.

Выгода, которую дает технология JSP в сравнении с другими веб-технологиями заключается в том, что JSP является платформонезависимой, переносимой и легко расширяемой технологией для разработки веб-приложений.

JavaServer Pages (JSP) позволяют отделить динамическую часть страниц от статического HTML. Процедура довольно проста, создаёте обычный код HTML (статический), а динамическую часть заключаете в специальные теги “<%>”.

JSP страницы имеют расширение .jsp и размещаются там же, где и обычные Web страницы. Структура таких страниц может состоять из пяти конструкций: HTML, комментарии, скриптовые элементы, директивы и действия. JSP страница при компиляции преобразуется в обычный сервлет со статическим содержимым, которое направляется в поток вывода, связанный с методом service. Поэтому при первом запросе этот процесс может вызвать некую задержку, но в большинстве своём незаметную первому пользователю. Комментарии в документе или программе служат к объяснению содержимого. Они не являются причиной замедления программы, так как транслятор и исполнитель их игнорируют. Скриптовые элементы позволяют вам указать код на языке Java, который впоследствии станет частью конечного сервлета, директивы дадут вам возможность управлять всей структурой сервлета, а действия служат для задания существующих используемых компонентов, а также для контроля над поведением движка JSP. Для упрощения работы со скриптами имеются заранее определённые переменные, такие как request, response, pageContext, session, out, application, config, page, exception.

Действия JSP используют конструкции с синтаксисом XML для управления работой движка сервлета. Вы можете динамически подключать файл, многократно использовать компоненты JavaBeans, направить пользователя на другую страницу или сгенерировать HTML для Java plugin. Все эти действия детально рассмотрены далее. Помните что как и во всем XML, имена элементов и атрибутов регистрозависимы. Действия можно разделить на две группы: стандартные и созданные.

ActionBean представляет собой интерфейс для всех классов, которые отвечают на интерфейсных пользовательских событиях. Реализации получать информацию о событии (обычно отправки формы) двумя способами. Первый через объект ActionBeanContext которая всегда будет установлен на ActionBean до любого пользователя реализован "обработчик" методы, которые вызываются. Второй через установку свойств JavaBean стиля на объекте (и вложенных JavaBeans). Значения, представленные в HTML форме будет привязан к собственности на объект, если такое свойство определяется. ActionBeanContext используется в основном для обеспечения

доступа к Servlet API, например, запроса и ответа, и другую информацию, полученную в ходе предварительной обработки запроса [7].

Классы ActionBean расположены в директиве “src”. В основном, в папке “actions” в поддиректории папки “src”. Мы должны определить эту папку как папка класса ActionBean в конфигурационном файле проекте web.xml.

Моделью нашего класса является классы Model. Каждый класс определяет и делает маппинг на одну таблицу в базе данных. Атрибуты класса определяют столбцов в таблице. Классы Model расположены в папке проекта “models”. Мы использовали Hibernate для работы с Моделью и связи с базой данных.

### 3.3 Stripes

Stripes – открытый веб-фреймворк основанный на паттерне MVC. Целью проекта является создание более легковесного фреймворка, нежели Struts, с использованием таких технологий Java как аннотации и средства настраиваемой типизации (Java Generics), которые были введены в версии Java 1.5 для реализации принципа «convention over configuration». Основной акцент делается на том, что простые соглашения, используемые в данном фреймворке, уменьшают избыточность в настройке. Фактически это означает то, что приложения, созданные с использованием фреймворка Stripes, нуждаются всего лишь в настройке в большинстве случаев, что позволяет снизить накладные расходы связанные с разработкой и поддержкой программного продукта [2].

Основной движущей силой Stripes является то, что разработка веб-приложений в Java это просто слишком много работы. Похоже, что каждый существующие рамки требуется уйма конфигурации [8].

Основные преимущества:

- отсутствие внешней конфигурации для каждой страницы/действия (Action Bean-ы – автоматически обнаруживаются и настраиваются используя аннотации);
- мощный механизм привязки, которые будут строить сложные объекты из параметров запроса;
- простота в использовании (и локализованы) проверки и тип системы преобразования;
- система Локализации, которая работает по прямой ссылке JSP->JSP links;
- простота в использовании индексированного свойства поддержки;
- встроенная поддержка для нескольких событий в форме;
- прозрачные возможности загружать файл;
- аннотации заменяют XML файлы для конфигурации;
- jsp макеты;
- открытый исходный код;
- совместим с Google App Engine.

### 3.4 JavaScript

JavaScript – прототипно-ориентированный сценарный язык программирования. Java Script обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам.

Основные архитектурные черты: динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса.

На Java Script оказали влияние многие языки, при разработке была цель сделать язык похожим на Java, но при этом лёгким для использования непрограммистами [13]. Языком Java Script не владеет никакая компания или организация, что отличает его от ряда языков программирования, используемых в веб-разработке. Название «JavaScript» является зарегистрированным товарным знаком компании Oracle Corporation.

В статье «The World’s Most Misunderstood Programming Language Has Become the World’s Most Popular Programming Language» [9]. «Самый неправильно понятый язык программирования в мире стал самым популярным в мире языком программирования») Дуглас Крокфорд (англ.) утверждает, что лидирующую позицию Java Script занял в связи с развитием AJAX, поскольку браузер стал преобладающей системой доставки приложений. Он также констатирует растущую популярность Java Script, то, что этот язык встраивается в приложения, отмечает значимость языка.

Java Script является объектно-ориентированным языком, но используемое в языке прототипирование обуславливает отличия в работе с объектами по сравнению с традиционными класс-ориентированными языками. Кроме того, Java Script имеет ряд свойств, присущих функциональным языкам – функции как объекты первого класса, объекты как списки, карринг, анонимные функции, замыкания – что придаёт языку дополнительную гибкость.

Несмотря на схожесть с Си синтаксис, Java Script по сравнению с языком Си имеет коренные отличия:

- объекты, с возможностью интроспекции;
- функции как объекты первого класса;
- автоматическое приведение типов;
- автоматическая сборка мусора;
- анонимные функции.

Синтаксис языка Java Script во многом напоминает синтаксис Си и Java, семантически же язык гораздо ближе к Self, Smalltalk или даже Лиспу.

В JavaScript все идентификаторы регистра зависимы, в названиях переменных можно использовать буквы, подчёркивание, символ доллара, арабские цифры. Названия переменных не могут начинаться с цифры. А для оформления однострочных комментариев используются «//»,



многострочные и внутрискрочные комментарии начинаются «с /\*» и заканчиваются «\*/».

JavaScript используется в клиентской части веб-приложений: клиент-серверных программ, в котором клиентом выступает браузер, а сервером — веб-сервер, имеющих распределённую между сервером и клиентом логику. Обмен информацией в веб-приложениях происходит по сети. Одним из преимуществ такого подхода является тот факт, что клиенты не зависят от конкретной операционной системы пользователя, поэтому веб-приложения являются кроссплатформенными сервисами.

### 3.5 jQuery

jQuery – библиотека JavaScript, фокусирующаяся на взаимодействии JavaScript и HTML. Библиотека jQuery помогает легко получать доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими. Также библиотека jQuery предоставляет удобный API для работы с AJAX.

HTML был одной из первых вещей, которую Джон Резиг освоил, когда он только начал заниматься программированием. Резиг программировал на QBasic, когда один его знакомый показал ему, как создать веб-страницу (используя Angelfire), а также основы HTML. Отец подарил ему на Рождество две книги по HTML. Именно тогда, когда он только начал программировать на Visual Basic, HTML и веб-дизайн очень заинтересовали его.

Но страсть к Java Script пришла значительно позже, примерно в 2004 году. Тогда Резиг получал степень в области компьютерных наук и работал на полставки в местной фирме Brand Logic. Он занимался дизайном сайта, в котором создавался пользовательский скроллинг. Джон был разочарован и расстроен, особенно потому, что использовал код других разработчиков, после чего решил серьёзно изучить Java Script. Изучив, пришел к выводам, что Java Script – это простой, но изящный язык, который является невероятно мощным для решения многих задач. В течение ближайшей пары лет Джон создал множество различных Java Script-приложений, прежде чем закончить создание jQuery. Основной целью создания jQuery Резиг видел возможность закодировать многократно куски кода, которые позволяют упростить Java Script и использовать их так, чтобы не беспокоиться о кросс-браузерных вопросах. Библиотека была представлена общественности на компьютерной конференции «BarCamp» в Нью-Йорке в 2006 году, которая имеет возможности:

- движок кроссбраузерных CSS-селекторов Sizzle, выделившийся в отдельный проект;
- переход по дереву DOM, включая поддержку XPath как плагина;
- события;
- визуальные эффекты;
- AJAX-дополнения;

- java script-плагины.

Точно так же, как CSS отделяет визуализацию от структуры HTML, JQuery отделяет поведение от структуры HTML. Например, вместо прямого указания на обработчик события нажатия кнопки управление передается JQuery, которая идентифицирует кнопки и затем преобразовывает его в обработчик события клика. Такое разделение поведения и структуры также называется принципом ненавязчивого JavaScript.

Библиотека JQuery содержит функциональность, полезную для максимально широкого круга задач. Тем не менее, разработчиками библиотеки не ставилась задача совмещения в JQuery функций, которые подошли бы всюду, поскольку это привело бы к большому коду, большая часть которого не востребована. Поэтому была реализована архитектура компактного универсального ядра библиотеки и плагинов. Это позволяет собрать для ресурса именно ту JavaScript-функциональность, которая на нём была бы востребована.

### 3.6 iText

iText – это библиотека классов для генерации, анализа и изменения документов в форматах Portable Document Format (PDF), а также XML, HTML и RTF.

Начиная с версии 5 библиотека имеет двойную лицензию – AGPL (вынуждающую предоставлять пользователям полный исходный текст приложения) либо коммерческую. Однако исходный текст версии 4.2 доступен под лицензией MPL/LGPL.

Библиотека была разработана Бруно Лоуаги зимой 1998 года, как в проекта Гентского университета, чтобы создать приложение для PDF документов для администрации студента. Библиотека (ранее известный как rugPdf) была очень маленькой и сложной библиотекой с минимальной функциональностью, например, просто читать и писать PDF файлы. Разработчик должен был быть осведомлен синтаксисом PDF, объектами, операторами и операндами для работы с библиотекой.

В 1999 году Лоуаги выбросил код rugPdf и написал новую библиотеку - iText, с нуля. 14 февраля 2000 года Лоуаги создал iText как библиотеку, чтобы Java разработчики могли использовать его для создания PDF-документов без знания синтаксиса PDF, и выпустил его как свободный продукт с открытым исходным кодом. Летом 2000 года Пауло Соареш присоединился к проекту и сейчас является одним из главных разработчиков новых функций iText.

iText был портирован на .NET Framework с именем iTextSharp. iTextSharp написан в C# и имеет отдельный базовый код, но это синхронизируется с iText релизом [10].

Особенности, которой является:

- отправлять PDF на браузер;
- генерация динамических документов от XML файла или базы данных;

- использует много интерактивных функций PDF;
- добавляет закладки, номера страниц, штрих коды;
- объединяет и управляет PDF страницы;
- автоматическое заполнение PDF форм;
- добавляет цифровые подписи к PDF файлам.

### 3.7 CSS

Формальный язык описания внешнего вида документа, написанного с использованием языка разметки – Cascading Style Sheets (CSS - каскадные таблицы стилей).

Данный язык в основном используется как средство описания, оформления внешнего вида веб-страниц, написанных с помощью языков разметки HTML и XHTML, но может также применяться к любым XML-документам, например, к SVG или XUL.

Для задания цветов, шрифтов, расположения отдельных блоков и других аспектов представления внешнего вида веб-страниц используется каскадные таблицы стилей CSS. Разделение описания логической структуры веб-страницы (которое производится с помощью HTML или других языков разметки) от описания внешнего вида этой веб-страницы (которое теперь производится с помощью формального языка CSS) является основной целью разработки CSS. Такое преимущество может предоставить большую гибкость и возможность управления его представлением, увеличить доступность документа, а также уменьшить сложность и повторяемость в структурном содержимом. Еще одно преимущество языка CSS позволяет представлять один и тот же документ в различных стилях или методах вывода, таких как экранное представление, печатное представление, чтение голосом (специальным голосовым браузером или программой чтения с экрана), или при выводе устройствами, использующими шрифт Брайля.

Таблицы стилей содержат в себе правила CSS и располагаются в таблицах стилей, поэтому правила CSS пишутся на формальном языке CSS. Выше сказанные таблицы стилей которые описывают внешний вид располагаются в самом веб-документе и они описывают внешний вид, а которые имеют формат CSS (на самом деле формат CSS – это обычный текстовый файл. В файле с форматом .css содержится перечень правил CSS и комментарии).

Когда таблица стилей находится в отдельном файле таблицы стилей могут быть подключены и внедрены четырьмя различными способами в описываемый веб-документ. С помощью тега `<link>` расположенном между открытым тегом `<head>` и закрытым тегом `</head>` может быть подключена веб-документу (открытый тег `<link>` имеет атрибут `href`, имеющий значение адрес этой таблицы стилей). Все правила данной таблицы действуют на протяжении всего документа.

Известно что HTML-документы может представлена в древовидной форме, на основании иерархии элементов. Элементы HTML друг для друга могут быть родительскими, дочерними, элементами-предками, элементами-потомками, сестринскими.

Если в иерархической структуре документа элемент находится сразу, непосредственно над этим элементом, то данный элемент является родителем другого элемента. А если элемент является предком другого элемента, то в иерархической структуре документа он находится где-то выше этого элемента.

Допустим, в документе присутствуют два абзаца `p`, включающие в себя шрифт с полужирным начертанием `b`. Тогда элементы `b` будут дочерними элементами своих родительских элементов `p`, и потомками своих предков `body`. В свою очередь, для элементов `p` элемент `body` будет являться только родителем. И кроме того, эти два элемента `p` будут являться сестринскими элементами, как имеющими одного и того же родителя – `body`.

При помощи селекторов в CSS могут задаваться кроме одиночных элементов, но и элементы, являющиеся потомками, дочерними или сестринскими элементами других элементов.

CSS – одна из широкого спектра технологий, одобренных консорциумом W3C и получивших общее название «стандарты Web» [8]. В 1990-х годах стала ясна необходимость стандартизировать Web, создать какие-то единые правила, по которым программисты и веб-дизайнеры проектировали бы сайты. Так появились языки HTML 4.01 и XHTML, и стандарт CSS.

В начале 1990-х различные браузеры имели свои стили для отображения веб страниц. HTML развивался очень быстро и был способен удовлетворить все существовавшие на тот момент потребности по оформлению информации, поэтому CSS не получил тогда широкого признания. Хокон Виум Ли в 1994 году был предложен термин «каскадные таблицы стилей». Позже вместе с Бертом Босом он стал развивать CSS.

В отличие от многих существовавших на тот момент языков стиля, CSS использует наследование от родителя к потомку, поэтому разработчик может определить разные стили, основываясь на уже определенных ранее стилях. В середине 1990-х Консорциум Всемирной паутины (W3C) стал проявлять интерес к CSS, и в декабре 1996 года была издана рекомендация CSS1.

### 3.8 Hibernate

Библиотека для языка программирования Java, предназначенная для решения задач объектно-реляционного отображения (object-relational mapping – ORM) это Hibernate. Hibernate представляет собой программное обеспечение распространяемое на условиях GNU Lesser General Public License с открытым исходным кодом (open source). Библиотека для языка программирования Java,

представляет собой каркас для показа объектно-ориентированной модели данных в традиционные реляционные базы данных [2].

Освобождение разработчика от максимального объема низкоуровневого программирования по обеспечению хранения объектов в реляционной базе данных является целью Hibernate. Инженер может использовать Hibernate не только в процессе проектирования системы классов, таблиц «с нуля», но и может работать с уже созданной базой данных.

Hibernate выполняет следующие задачи:

- выполняет связи классов Java с таблицами базы данных;
- выполняет типы данных Java с типами данных SQL;
- представляет средства для автоматической генерации;
- обновляет набор таблиц;
- строит запросы и обрабатывает полученные данные;
- уменьшает время разработки написанию кода SQL и JDBC;

Обычно разработчик обрабатывает результаты набора данных и преобразования объектов в ручную, а Hibernate облегчает эту задачу, тем, что он автоматизирует генерацию запросов SQL и легче делается запрос программы (портирование) на любые базы данных SQL.

Чтобы написать следующую функцию Hibernate, дадим сначала определение что такое «POJO». «POJO» – это стандартные Java-объекты. Для «POJO» Hibernate обеспечивает прозрачную поддержку сохранности данных (persistence). Наличие конструктора по умолчанию без параметров это единственное обязательное требование для сохраняемого класса. Чтобы в приложениях было нормальное поведение, можно использовать такие два метода как equals() и hashCode().

С помощью конфигурационных XML файлов или Java-аннотаций осуществляется маппинг (проецирование, отображение) Java-классов с таблицами баз данных. Hibernate может вызывать исходный код для классов длительного хранения при применении файла XML. Если применяется аннотация, то можно и не генерировать исходный код для классов длительного хранения. Для поддержки схемы базы данных библиотека может применить как файл XML, также и аннотацию.

В базе данных есть такие отношения как «один-ко-многим» и «многие-ко-многим». В Hibernate есть возможность по организации этих отношений между классами. Есть еще рефлексивные отношения, то есть когда объект имеет связь «один-ко-многим» с другими экземплярами своего собственного типа данных. Hibernate может управлять этими отношениями.

Библиотека поддерживает отображение пользовательских типов значения. Это делает возможным сценарии, такие как переопределяют тип по умолчанию SQL. При отображении столбца Hibernate выбирает свойства. Выполняет копию перечисляемого типа Java на поле БД, будто они являются обычными свойствами. А именно копирует одно свойство в несколько колонок.

Набор (Set) и список (List) это коллекции объектов данных и хранятся в виде коллекций Java-объектов. В Java 5 введены общие классы (Generics). Библиотека может настроенным быть на отложенные загрузки. Начиная с Hibernate «ленивые» загрузки могут быть вариантом по умолчанию.

На каскадные операции настроены связанные объекты. Приведём пример, допустим музыкальный альбом – Album это родительский класс, может быть настроен на каскадное сохранение и/или удаление своего отпрыска Track. Каскадные операции значительно сокращают время разработки и обеспечивает сохранность. Выполняя SQL обновление только при изменении полей персистентных объектов, функция проверки изменения данных (dirty checking) позволяет избежать ненужной записи действий в базу данных.

Широкое распространение Hibernate привело JCP к разработке спецификации JDO. JDO является стандартной технологией ORM на платформе JavaEE. Библиотека Hibernate совместима с JSR-220/317 предлагает стандартные средства JPA.

Hibernate используется не только как отдельное приложение Java, но и устанавливается в приложениях JavaEE, которые выполняются на сервере (компонент EJB или сервлет). Еще Hibernate используется как дополнительная возможность подключаться к языкам программирования. Например, Adobe интегрировал Hibernate в девятую версию ColdFusion (запускаемый на серверах с поддержкой приложений J2EE) с уровнем абстракции новых функций и синтаксиса, приложенных к CFML.

### 3.9 Веб-сервер Apache Tomcat

Веб-сервер – сервер, принимающий HTTP-запросы от клиентов, обычно веб-браузеров, и выдающий им HTTP-ответы, как правило, вместе с HTML-страницей, изображением, файлом, медиа-поток или другими данными [11].

Веб-сервером называют как программное обеспечение, выполняющее функции веб-сервера, так и непосредственно компьютер, на котором это программное обеспечение работает.

Клиент, которым обычно является веб-браузер, передаёт веб-серверу запросы на получение ресурсов, обозначенных URL-адресами. Ресурсы – это HTML-страницы, изображения, файлы, медиа-поток или другие данные, которые необходимы клиенту. В ответ веб-сервер передаёт клиенту запрошенные данные. Этот обмен происходит по протоколу HTTP.

Apache Tomcat контейнер сервлетов с открытым исходным кодом, разрабатываемый Apache Software Foundation. Реализует спецификацию сервлетов и спецификацию JavaServer Pages (JSP) и JavaServer Faces (JSF). Написан на языке Java [2].

Tomcat позволяет запускать веб-приложения, содержит ряд программ для самоконфигурирования.

Tomcat используется в качестве самостоятельного веб-сервера, в качестве сервера контента в сочетании с веб-сервером Apache HTTP Server, а также в качестве контейнера сервлетов в серверах приложений JBoss и GlassFish.

Разработка и поддержка Tomcat осуществляется фондом Apache Software Foundation и добровольцами. Пользователи имеют свободный доступ к исходным кодам и бинарным файлам Tomcat согласно лицензии Apache License 2.0. Номера версий Tomcat начинаются с 3.0.x (предыдущие версии были выпущены Sun для внутреннего пользования).

Catalina – контейнер сервлетов Tomcat'a. Catalina реализует спецификацию сервлетов и JavaServer Pages (JSP). Архитектором Catalina являлся Craig McClanahan.

Coyote – компонент стека HTTP Tomcat'a, который поддерживает протокол HTTP 1.1 для веб серверов или контейнера приложений. Coyote прослушивает входящие соединения на определённом TCP порту сервера и пересылает запросы в механизм Tomcat для обработки запросов и отправляет ответ назад запрашивающему клиенту.

Jasper – механизм JSP Tomcat'a. Tomcat 5.x использует Jasper 2, который является реализацией спецификации JavaServer Pages 2.0 Sun Microsystems. Jasper анализирует JSP-файлы, чтобы компилировать их в Java код, как сервлеты (которые могут быть обработаны с помощью Catalina). Во время выполнения, Jasper может автоматически обнаруживать изменения JSP-файла и перекомпилировать его.

## 4 Экономическая часть

### 4.1 Трудовые ресурсы, используемые в работе

На данном этапе проанализируем экономическую часть реализации проекта, отражающую временные, трудовые и финансовые затраты на проект.

На данном этапе мы проанализируем экономическую часть реализации проекта, отражающую временные, трудовые и финансовые затраты на проект.

В данном дипломном проекте задействованы:

- разработчик;
- руководитель проекта.

Количество сотрудников, задействованных в разработке ПП, представлено в таблице 4.1

Таблица 4.1 – Данные о работниках, задействованных в проекте и их заработная плата

| Наименование         | Количество | Зарботная плата, тенге |
|----------------------|------------|------------------------|
| Разработчик          | 1          | 120000                 |
| Руководитель проекта | 1          | 130000                 |
| Итого                | 2          | 150000                 |

### 4.2 Оборудование, используемое в работе

Оборудование, используемое при разработке ПП, представлено в таблице 4. 2. Цены на оборудование приведены без учета НДС.

Таблица 4.2 – Перечень оборудования, необходимого для разработки ПП

| Наименование           | Характеристики  | Количество единиц | Цена за единицу, тенге | Общая сумма, тенге |
|------------------------|---|-------------------|------------------------|--------------------|
| Персональный компьютер | Ноутбук Acer V5-571PG-323c4G50Mass (15.6" Multi Touch, Intel Core i3) | 1                 | 105000                 | 105000             |
| Модем                  | USB - модем от ALTEL 4G (LTE) с пакетом «Гранд Премьера» (ZTE MF823)  | 1                 | 7500                   | 7500               |
| Всего                  |   |                   |                        | 112500             |



#### 4.3 Программное обеспечение, используемое в работе

Программное обеспечение, используемое в работе, представлено в таблице 4.3. Цены на ПО приведены без учета НДС.

Таблица 4.3 – Перечень программного обеспечения, используемого при разработке программного продукта

| Программное обеспечение  | Стоимость, тенге |
|--|------------------|
| Операционная система Microsoft Windows 7 Professional, 64 bit, English SP1 | 28750            |
| IntelliJ Idea 13 Enterprise Edition  | 103000           |
| Apache Tomcat 7, Хостинг и домен   | 63000            |
| Итого  | 194750           |

#### 4.4 Сроки реализации проекта

Проектирование и разработка программного продукта состоит из этапов и включает следующие виды работ:

- первый этап – постановка задачи, сбор необходимой информации, разработка структуры базы данных;
- второй этап – создание объектов конфигурации и их форм;
- третий этап – проектирование клиентского интерфейса для программного продукта;
- четвертый этап – создание ролей, пользователей и разработка интерфейсов для них;
- пятый этап – загрузка системы на хостинг;
- шестой этап – проверка на работоспособность ПО.

График разработки проекта представлен в таблице 4.4.

Таблица 4.4 – Этапы и сроки реализации проекта

| Наименование этапа |   | Недели от начала работ |   |   |   |   |   |   |   |   |  |
|--------------------|---|------------------------|---|---|---|---|---|---|---|---|--|
|                    |   | 1                      | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |  |
| 1 этап             | Постановка задачи   |                        |   |   |   |   |   |   |   |   |  |
|                    | Подбор и изучение литературы                                |                        |   |   |   |   |   |   |   |   |  |
|                    | Разработка структуры БД                                     |                        |   |   |   |   |   |   |   |   |  |
| 2 этап             | Создание объектов конфигурации, их форм                     |                        |   |   |   |   |   |   |   |   |  |
| 3 этап             | Заполнение базы данных                                      |                        |   |   |   |   |   |   |   |   |  |
| 4 этап             | Создание ролей, пользователей<br>Разработка интерфейсов для |                        |   |   |   |   |   |   |   |   |  |

|        |                             |  |  |  |  |  |  |  |  |  |  |
|--------|-----------------------------|--|--|--|--|--|--|--|--|--|--|
|        | пользователей               |  |  |  |  |  |  |  |  |  |  |
| 5 этап | Загрузка системы на хостинг |  |  |  |  |  |  |  |  |  |  |
| 6 этап | Тестирование ПО             |  |  |  |  |  |  |  |  |  |  |
|        | Отладка ПО                  |  |  |  |  |  |  |  |  |  |  |

#### 4.5. Затраты на разработку системы

Затраты на разработку системы (производственные затраты) представляют собой единовременные расходы на всех этапах инновационного процесса: исследование, разработка, опытная проверка. Определение затрат на разработку проекта производится путем составления калькуляции плановой себестоимости. В плановую себестоимость включают все затраты, связанные с ее выполнением, независимо от источника их финансирования.

Стоимость разработки проекта определяется по формуле:

$$Сб = ФОТ + О_{сн} + А + Э + Н, \quad (4.1)$$

где Сб – себестоимость;  
 ФОТ – фонд оплаты труда;  
 О<sub>сн</sub> – социальный налог;  
 А – амортизационные отчисления;  
 Э – затраты на электроэнергию;  
 Н – накладные расходы.

#### 4.6 Расчет фонда оплаты труда

Фонд оплаты труда (ФОТ) – это суммарные издержки предприятия на оплату труда всех работников за определенный период, который формируется из основной и дополнительной заработной платы и определяется по формуле:

$$ФОТ = З_{осн} + З_{доп}, \quad (4.2)$$

где З<sub>осн</sub> - основная заработная плата;  
 З<sub>доп</sub> – дополнительная заработная плата.

На этапах разработки ПП, участники разработки задействованы неравноценно, для этого необходимо рассчитать заработную плату за один час работы, а затем общий размер заработной платы. Заработная плата за один час рассчитывается по формуле:

$$Н = ЗП_{М} / Д_{р} * Ч_{р}, \quad (4.3)$$

где ЗП<sub>М</sub> – ежемесячный размер заработной платы, тенге;

Др – количество рабочих дней в месяце (24 дня – семидневная рабочая неделя);

Чр – продолжительность рабочего дня, час (при 8-часовом рабочем дне).

Разработчик:  $H=120000/24*8=625$  тенге/час

Руководитель:  $H=130000/24*8=677$  тенге/час.

Длительность цикла в днях по каждому виду работ определяется по формуле:

$$t_n = T/q_n * z * K, \quad (4.4)$$

где Т – трудоемкость этапа, норма-час;

$q_n$  – количество исполнителей по этапу;

z – продолжительность рабочего дня, z= 8 часов;

K – коэффициент выполнения норм времени, K = 1,1.

Полученную величину  $t_n$  округляю в большую сторону до целых дней.

Сводные результаты расчета затрат на основную заработную плату работников, задействованных в разработке ПП представлены в таблице 4.5.

Таблица 4.5 – Сводные результаты расчета затрат на основную заработную плату

| Наименование работ                      | Исполнитель  | Трудоемкость |                         | Длительность цикла, днях | Зарботная плата за час работы, тенге | Сумма зарплаты, тенге |
|---|--------------|--------------|-------------------------|--------------------------|--------------------------------------|-----------------------|
|   |              | Нормо-час    | % от общей трудоемкости |                          |                                      |                       |
| Постановка задачи                       | Руководитель | 32           | 7,84                    | 4                        | 677                                  | 18956                 |
| Подбор и изучение литературы            | Руководитель | 24           | 5,88                    | 3                        | 677                                  | 16248                 |
|   | Разработчик  | 24           | 5,88                    | 3                        | 625                                  | 15000                 |
| Разработка структуры БД                 | Разработчик  | 32           | 7,84                    | 4                        | 625                                  | 20000                 |
| Создание объектов конфигурации, их форм | Разработчик  | 80           | 19,61                   | 10                       | 625                                  | 50000                 |

|                                   |             |    |      |   |     |       |
|-----------------------------------|-------------|----|------|---|-----|-------|
| Разработка клиентского интерфейса | Разработчик | 24 | 5,88 | 3 | 625 | 15000 |
| Создание ролей, пользователей     | Разработчик | 8  | 1,96 | 1 | 625 | 5000  |

Продолжение таблицы 4.5

|                             |              |     |       |    |     |        |
|-----------------------------|--------------|-----|-------|----|-----|--------|
| Загрузка системы на хостинг | Разработчик  | 8   | 4,1   | 1  | 625 | 5000   |
| Тестирование ПО             | Разработчик  | 24  | 5,88  | 3  | 625 | 15000  |
| Отладка ПО                  | Разработчик  | 24  | 5,88  | 3  | 625 | 15000  |
| Проверка и сдача отчета     | Разработчик  | 8   | 1,96  | 1  | 625 | 5000   |
|                             | Руководитель | 8   | 1,96  | 1  | 677 | 5416   |
| Итого                       |              | 344 | 100,0 | 43 |     | 215620 |

Дополнительная заработная плата составляет 10 % от основной заработной платы и рассчитывается по формуле 4.6:

$$Z_{\text{доп}} = Z_{\text{осн}} * 0,1 \text{ тенге.} \quad (4.5)$$

$$Z_{\text{доп}} = 215\,620 * 0,1 = 21\,562 \text{ тенге.}$$

Таким образом, суммарный фонд оплаты труда составит:

$$\text{ФОТ} = 215620 + 21562 = 237182 \text{ тенге.}$$

#### 4.7 Расчет затрат по социальному налогу

Социальный налог составляет 11% (ст. 358 п.1 НК РК) от дохода работника, и рассчитывается по формуле:

$$C_{\text{н}} = (\text{ФОТ} - \text{ПО}) * 0,11, \quad (4.6)$$

где ПО – пенсионные отчисления, которые составляют 10% от ФОТ и социальным налогом не облагаются:

$$\text{ПО} = \text{ФОТ} * 0,1 \text{ тенге.} \quad (4.7)$$

$$\text{ПО} = 237182 * 0,1 = 23718,2 \text{ тенге.}$$

Размер отчислений на социальные нужды составит:

$$C_n = (237182 - 23718,2) * 0,11 = 23481,02 \text{ тенге.}$$

#### 4.8 Расчет амортизационных отчислений

Амортизационные отчисления рассчитываются по формуле:

$$A_i = N_A * C_{\text{пер}}, \quad (4.8)$$

где  $N_A$  – норма амортизации;

$C_{\text{пер}}$  – первоначальная стоимость оборудования.

Норма амортизации  $N_A$  на компьютерную технику и на программное обеспечение составляет 40% от всей стоимости. Амортизационные отчисления по используемому оборудованию и программному обеспечению составят:

$$A_1 = 0,4 * 105000 = 42000 \text{ тенге.}$$

$$A_2 = 0,4 * 7500 = 3000 \text{ тенге.}$$

$$A_3 = 0,4 * 28750 = 11500 \text{ тенге.}$$

$$A_4 = 0,4 * 103\,000 = 41200 \text{ тенге.}$$

$$A_5 = 0,4 * 63000 = 25200 \text{ тенге.}$$

$$A = 42000 + 3000 + 11500 + 41200 + 25200 = 122,900 \text{ тенге.}$$

Расчет суммы амортизации за 1 месяц:

$$A = 122900 / 12 = 10241,66 \text{ тенге.}$$

Сумма амортизации за 2 месяца равна:

$$A = 10241,66 * 2 = 20483,33 \text{ тенге.}$$

#### 4.9 Расчет затрат на электроэнергию и накладных расходов

Так как, в процессе производства используется электрооборудование необходимо рассчитать затраты на электроэнергию. Затраты на электроэнергию для производственных нужд включают в себя расходы электроэнергии на оборудование и дополнительные нужды.

$$\mathcal{E} = \mathcal{E}_{\text{эл.эн.обор}} + \mathcal{E}_{\text{доп.нуж}}, \quad (4.9)$$

где  $Z_{\text{эл.эн.обор}}$  – затраты на электроэнергию оборудования;

$Z_{\text{доп.нуж}}$  – затраты электроэнергии на дополнительные нужды.

Расходы электроэнергии на оборудование рассчитываются по формуле:

$$Z_{\text{эл.эн.обор}} = W * T * S * K_{\text{исп}}, \quad (4.10)$$

где  $W$  - потребляемая мощность, Вт;

$T$  – количество часов работы оборудования;

$S$  – стоимость киловатт-часа электроэнергии (1кВтч = 17 тенге);

$K_{\text{исп}}$  - коэффициент использования ( $K_{\text{исп}} = 0,9$ );

$W = 70 \text{ Вт} = 0,07 \text{ кВт}$  (Мощность персонального компьютера);

$S = 17 \text{ тенге}$ .

Сумма затрат на электроэнергию основного оборудования составляет:

$$Z_{\text{эл.эн.обор}} = 0,07 * 344 * 17 * 0,9 = 368,42 \text{ тенге.}$$

Затраты на дополнительные нужды берутся по укрупненному показателю в размере 5% от затрат на оборудование:

$$Z_{\text{доп.нуж}} = 0,05 * Z_{\text{эл.эн.обор}} \quad (4.11)$$

$$Z_{\text{доп.нуж}} = 0,05 * 368,42 = 18,42 \text{ тенге.}$$

Суммарные затраты на электроэнергию составляют:

$$\mathcal{E} = 368,42 + 18,42 = 368,84 \text{ тенге.}$$

Расходы на интернет ( $P_{\text{и}}$ ) составляют 4200 тенге в месяц. Итого за 3 месяца  $P_{\text{и}} = 12600$  (тенге).

Таблица 4.6 - Расчет расхода иных материалов

| Наименование                       | Количество, шт. | Цена за единицу, тг. | Сумма |
|------------------------------------|-----------------|----------------------|-------|
| Лицензирование                     | 1               | 17000                | 17000 |
| Картридж                           | 1               | 1000                 | 1000  |
| Съемный носитель<br>Transcend 2 Gb | 1               | 1200                 | 1200  |
| Бумага                             | 1               | 850                  | 850   |
| Маркеры                            | 1               | 150                  | 150   |
| Диск CD                            | 100             | 30                   | 3000  |
| Итого                              |                 |                      | 23200 |

$$P_{\text{пр}} = (\text{ФОТ} + C_{\text{н}} + A + M + P_{\text{и}} + \text{Э}) * 0,2, \quad (4.12)$$

где М - расходные материалы (бумага, картриджи, канцелярские принадлежности и другие). Поэтому определяют дополнительную статью затрат – прочие производственные расходы (Рпр). Эти расходы представляют 20% от всех затрат.

$$P_{\text{пр}} = (237182 + 23\,481,018 + 20\,483,3333 + 23200 + 12600 + 368,8452) * 0,2 = 63463,03 \text{ тенге.}$$

Накладные расходы составляют 70% от всех затрат и рассчитываются по формуле:

$$НР = (\text{ФОТ} + C_{\text{н}} + A + \text{Э}) * 0,7 \quad (4.13)$$

Накладные расходы согласно формуле (4.10) составляют:

$$НР = (237182 + 23481,02 + 20483,33 + 368,84) * 0,7 = 197060,63 \text{ тенге}$$

Таким образом, в соответствии с формулой 1, суммарные затраты по разработке программного продукта составляют:

$$C_{\text{б}} = 237182 + 23481,02 + 20483,33 + 368,84 = 281515,19 \text{ тенге}$$

Смета затрат по разработке ПП и структура затрат представлены в таблице 4.7

Таблица 4.7 – Стоимость разработки автоматизированной системы управления конференциями

| Наименование статей затрат | Сумма, тенге | Доля в общей себестоимости, % |
|----------------------------|--------------|-------------------------------|
| ФОТ                        | 237182       | 49,1                          |
| Социальный налог           | 23481,02     | 4                             |
| Амортизационные отчисления | 20483,33     | 3                             |
| Затраты на электроэнергию  | 368,84       | 2                             |
| Накладные расходы          | 197060,63    | 41,2                          |
| Расходные материалы        | 23200        | 0,7                           |
| Итого                      | 501,75       | 100                           |

Диаграмма процентной составляющей статей затрат от общей себестоимости ПП представлена на рисунке 4.1



Рисунок 4.1 - Процентная составляющая статей затрат от общей себестоимости ПП

#### 4.10 Цена реализации

Цена реализации программного продукта складывается из его стоимости и прибыли:

$$Ц = Сб + П, \quad (4.14)$$

где Сб – стоимость продукта;

П – прибыль.

При определении первоначальной цены следует задать уровень рентабельности (20%) для реализации программного продукта:

$$Ц_{п} = Сб * (1+P/100), \quad (4.15)$$

где P – рентабельность (20%).

$$Ц_{п} = 501,75 * (1+0,2) = 602,13 \text{ тенге.}$$

Цена реализации готовой продукции рассчитывается по формуле:

$$Ц_{р} = Ц_{п} + НДС, \quad (4.16)$$

где НДС – налог на добавочную стоимость.

НДС рассчитывается по формуле:



$$\text{НДС} = \text{Ц}_\text{н} * 0,12 \quad (4.17)$$

$$\text{НДС} = 602,13 * 0,12 = 722,55 \text{ тенге.}$$

$$\text{Ц}_\text{р} = 602,13 + 722,55 = 674,38 \text{ тенге.}$$

Интегрированная система, которая обеспечивает преимущества для всех заинтересованных сторон: студентов, преподавателей и владельцев конференции. Система хранит доклады, статьи и научные исследования в одном месте.

Система управления конференциями это веб-приложение, которое поддерживает организацию конференций, в особенности научные конференции. Но наша система не будет только для проведения конференций, но и для проведения различных онлайн соревнований и хранения статьи преподавателей и оказание помощи в выборе руководителя проектов и дипломных работ.

В результате экономического расчета затраты на разработку программного продукта составили 501,75 тенге. Основной статьей расходов является заработная плата, которая составляет 49,1% от всех затрат. На втором месте накладные расходы, которые составляют 41,2% от всех затрат. Цена реализации программного продукта 674,38 – тенге.

## **5 Безопасность жизнедеятельности**

### **5.1 Характеристика условий труда программиста**

С развитием науки и техники у людей играет важную роль возможность безопасного выполнения своих должностных обязанностей. В связи с этим развивается наука о безопасности труда и жизни человека.

Здоровье и безопасность - комплекс мер, направленных на обеспечение безопасности человека в окружающей среде и сохранить его здоровым. Наука о здоровье и безопасности – это развитие методов и средств защиты от снижения уровня воздействия вредных и опасных фактор ниже допустимых значений. Развития мер ограничения ущерб в лицо ликвидации последствий чрезвычайных ситуаций в мирное и военное время.

Цели и содержание безопасности труда человека:

- выявление и изучение экологических факторов, которые отрицательно влияют на здоровье человека;
- ослабление и ограничение факторов для безопасности или исключить их, если это возможно;
- ликвидация последствий катастроф и стихийных бедствий.

Спектр практических задач безопасности труда человека, в первую очередь, которая основана выбором принципов защиты человека, для развития рационального использования ресурсов и охране человеческой среды от антропогенных источников и стихийных бедствий, а также это является инструментом, которая обеспечивает комфортное условие для жизни состоянии человека.

Защита здоровья работников, безопасности труда, ликвидации травматизма и профессиональных заболеваний является одним из главных качеств забот человеческого общества. Особое внимание обращено на необходимость эффективных форм организации труда, минимальные затраты на ручную работу, неквалифицированного труда, создание условий предотвращения профессиональных заболеваний и производственного травматизма.

На рабочем месте должны быть предоставлены условия защиты от возможных оказании действию опасных и вредных производственных факторов. Уровни этих факторов не должен превышать указанных правовых пределов и санитарно-технических норм.

Эти правила требуют создания рабочих мест по условиям труда, при которых воздействие опасных и вредных факторов производство труда либо полностью отменены, или находится в допустимых пределах.

В данном разделе дипломного проекта рассматриваются следующие вопросы:

- определение оптимальных условий труда инженера - программиста;
- расчет освещения;
- расчет степени шума.

Научно-технический прогресс внес значительные изменения в условиях производственной деятельности работников умственного труда. Их работа стала более напряженной, стрессовой, требующие значительных инвестиций, умственной, эмоциональной и физической энергии. Для этого потребовалось решить вопросы эргономики, гигиены и организации труда, нормативно-правовых режимов труда и отдыха.

В настоящее время компьютерные технологии широко используются во всех сферах человеческой деятельности. Человек подвергается ряду опасных и вредных производственных факторов при работе с компьютером: электромагнитные поля при радиочастотном диапазоне: ВЧ, УВЧ и СВЧ, инфракрасного и ионизирующих излучений, шума и вибрации, статического электричества.

Большое воздействие оказывает работа с компьютером на работающего программиста, а именно значительным психическим стрессом и нервно-эмоциональной нагрузкой, высокая интенсивность зрительной работы и достаточно большую нагрузку на мышцы рук при использовании клавиатуры компьютера. Важность сказывается на рациональной конструкции и компоновки рабочего места для поддержания оптимальной рабочей позы человека-программиста.

Особо важно сохранять в процессе работы с компьютером правильный режим труда и отдыха. При не соблюдении правил можно ожидать у программиста значительную низко способность работы с появлением напряжения зрительного аппарата, жалобы головной боли, раздражительность, нарушение сна, усталость, а также боль в глазах, в спине, шее и руках.

## 5.2 Окраска и коэффициенты отражения

Окраска кабинета и мебель должны помочь в создании благоприятных условий для визуального восприятия и для хорошего состояния работающего.

Светящийся объект, излучающий энергию в световом диапазоне, которые придают отражения от поверхности экрана, существенно влияют на точность знаков и влечет за собой вмешательство физиологического характера, которые приводят значительным стрессом, особенно при длительной работе в помещении. Отражение, так и отражения от вторичного источника света должны, достигнуты минимальных пределов. Для защиты от чрезмерной яркости могут быть применены занавесы и шторы.

Относительно окон рекомендуется следующая покраска стен и полов:

а) окна направленные на юг: - стены зеленовато-голубого или светло-голубого цвета; пол - зеленый;

б) окна направленные на север: - стены светло-оранжевого или оранжево-желтого цвета; пол - красновато-оранжевый;

в) окна направленные на восток: - стены желто-зеленого цвета; пол зеленый или красновато-оранжевый;

д) окна направленные на запад: - стены желто-зеленого или голубовато-зеленого цвета; пол зеленый или красновато-оранжевый.

В районах, где есть компьютер, необходимо предоставить следующие значения коэффициента отражения для потолка: 60 ... 70%, для стен: 40 ... 50%, для пола: около 30%. Для других плоскостей и рабочей мебели: 30 ... 40%.

Правильно разработанные и выполненные промышленное освещение улучшает условия зрительной работы, снижает утомляемость, повышает производительность труда, благотворно влияют на производственную среду, оказывая положительное психологическое воздействие на работника, повышает безопасность и снижает риск травм.

Недостаток света вызывает зрительное напряжение, снижает внимание, приводит к преждевременному наступлению усталости. Чрезмерно яркое освещение вызывает бликов, тем самым оказывая раздражение и боль в глазах. Искажение направление света на рабочем месте могут создать резкие тени, отражения, дезориентировать работника. Все эти факторы могут привести к аварии или профилактическому заболеванию, поэтому очень важен правильный расчет параметров освещения. Есть три типа освещения - естественное, искусственное и смешанное (искусственное и естественное вместе).

Естественное освещение – дневное освещение комнаты через окна, где свет проникает через проемы с наружной стороны. Естественное освещение, отличается тем, что широко варьируется в зависимости от времени суток, времени года, локальной области и других факторов.

Искусственное освещение используется при работе в темноте и днем, когда вы не можете предоставить нормированные значения коэффициента естественного освещения (облачная погода, короткий световой день). Комбинированное освещение – это освещение недостающее нормам естественное освещение, которое дополняется искусственным.

Искусственное освещение делится на рабочее, аварийное, эвакуационное, и безопасное. Рабочее освещение может быть общим или в сочетании естественного и искусственного освещения. Общее освещения это когда светильники располагаются в верхней зоне помещения равномерно или в отношении расположения оборудования. Комбинированное освещение повышает общее локальное освещение, так как добавляет к общему местное освещение.

Согласно СНиП II-4-79 в вычислительных центрах помещение должны применяться в комбинированной системе освещения.

При проведении визуальной работы категории высокой точности (минимальный размер объекта дискриминации на 0,3 ... 0,5 мм). Коэффициент естественного освещения (КЕО) не должна быть ниже 1,5%, в то время как средняя точность зрительных работ (минимальный размер 0.5 объектом

дискриминации ... 1,0 мм) КЕО должен составлять не ниже 1.0%. Вместо источников искусственного освещения широко применяют люминесцентные лампы типа ЛБ или ДРЛ, которые объединившись в светильники должны быть помещены над рабочей поверхности равномерно.

Требования освещения света в помещениях, где компьютеры устанавливаются следующим образом: зрительная работа при высокой точности составляет 300 лк освещения, при комбинированной 750 лк; соответственно при работе средней класса точности требуется от 200 до 300 лк.

Основным гигиеническим требованием является равномерное освещение всего поле зрения работающего. Другими словами, степень освещенности помещения и яркость экрана компьютера должны быть примерно равными, потому что яркий свет в периферийной зоне повышает интенсивность напряжения глаз и, как следствие, приводит к их усталости.

### 5.3 Параметры микроклимата

Параметры микроклимата могут сильно различаться в больших пределах, в то время как необходимое условие человеческой деятельности, чтобы поддерживать постоянную температуру тела через терморегуляции - это способность организма регулировать теплообмен с окружающей средой. Принцип регулирования микроклимата - создать оптимальные условия для тепла человеческого тела с окружающей средой.

Вычислительная машина - это существенный источник тепла, которое может привести к повышению температуры и снижению влажности в кабинете. В комнате, где есть компьютеры, особо важны некоторые параметры микроклимата. При санитарных нормах СН-245-71 при котором значение параметров микроклимата благоприятны для создания комфортной среды обитания работника. Эти нормы устанавливаются в зависимости от сезона, характера трудового процесса и характер производственных объектов приведены в таблице 5.1.

Объем кабинета, в которых размещены работники программисты составляют не меньше, чем 19.5 м<sup>3</sup>/человека с максимальным количеством одновременных работников в смену. Поставка норм свежего воздуха в кабинет, где размещены компьютеры, приведены в табл. 5.2.

Таблица 5.1 - Параметры микроклимата для кабинета, где установлен компьютер

| Сезон года | Параметр микроклимата          | Величина     |
|------------|--------------------------------|--------------|
| Холодный   | Температура воздуха в кабинете | 22...24°C    |
| Холодный   | Относительная влажность        | 40...60%     |
| Холодный   | Скорость движения воздуха      | до 0,1м/с    |
| Теплый     | Температура воздуха в кабинете | 23...25°C    |
| Теплый     | Относительная влажность        | 40...60%     |
| Теплый     | Скорость движения воздуха      | 0,1...0,2м/с |

Таблица 5.2 - Нормы подачи свежего воздуха в помещения, где расположены компьютеры

| Характеристика кабинета               | Объемный расход подаваемого в кабинет свежего воздуха, м <sup>3</sup> /на одного человека в час |
|---------------------------------------|---|
| Объем до 20м <sup>3</sup> на человека | Не менее 30   |
| 20...40м <sup>3</sup> на человека     | Не менее 20   |
| Более 40м <sup>3</sup> на человека    | Естественная вентиляция   |

Обеспечить комфортные условия организации работы в зависимости от сезона используют организационные методы (рациональной, чередование труда и отдыха) и оборудование (вентиляция, кондиционирование, системы отопления).

#### 5.4 Шум и вибрация

Шум, воздействуя на организм человека, снижает условие труда его работы. Работая долгое время при воздействии шума программист ощущает на себе раздражительность, головные боли, головокружение, частичную потерю памяти, утомляемость, потерю аппетита, и боли в ушах. Целый ряд таких нарушениях систем человеческого организма способствует к неблагоприятному эмоциональному состоянию человека до стрессового. Заострение внимания под влиянием сильного шума у работника наблюдается снижение, нарушение физиологических функций, появляется усталость из-за затрачивания энергии и умственных нагрузок, и ухудшение голос переключения. Все это сказывается на работоспособности программиста и на

его производительность, качество и безопасность работы. Длительное оказание действия интенсивного шума свыше 80 дБ (А) лицу, способствует к частичной или полной потере слуха [12].

В таблице 5.3 указаны в зависимости от категории тяжести труда и напряженности приближенные к уровню звука, являются безопасными по отношению сохранения здоровья и работы труда программиста.

Таблица 5.3 - Предельные уровни звука, дБ, на рабочих местах

| Категория напряженности труда | I. Низкая | II. Средняя | III. Высокая | IV. Очень высокая |
|-------------------------------|-----------|-------------|--------------|-------------------|
| Мало напряженный              | 80        | 80          | 75           | 75                |
| Умеренно напряженный          | 70        | 70          | 65           | 65                |
| Напряженный                   | 60        | 60          | -            | -                 |
| Очень напряженный             | 50        | 50          | -            | -                 |

На рабочем месте математиков, программистов уровень шума видеоматериала не должны быть выше - 50 дБА, и в вычислительных центрах обработки информации на компьютере – 65 дБА. Компьютеры выложенные противозумным материалом для понижения уровня шума стены и потолка в комнате. В вычислительных центрах степень вибрации можно понизить путем установки специального оборудования на виброизоляторах.

### 5.5 Электромагнитное и ионизирующее излучения

В настоящее время ученых считают, что краткосрочное и долгосрочное оказание всех видов излучений от экрана монитора не является опасным для здоровья персонала, обслуживающего вычислительную технику. Следует заметить, что взаимоисключающие данные о вреде опасности, связанных с излучением от экранов монитора, работающих на компьютерах, так и нет, и исследований в этой области продолжается.

В таблице 5.4 показаны в соответствии с СанПиН 2.2.2.542-96 допустимые значения параметров неионизирующих электромагнитных излучений от экрана компьютера.

Максимальный уровень рентгеновского излучения на рабочем месте оператора персонального компьютера, как правило, не превышает 10 мк Бар/ч, и интенсивность ультрафиолетового и инфракрасного излучений от экрана монитора в диапазоне 10 – 100 м Вт/м<sup>2</sup>.

Таблица 5.4 - Возможные значения параметров неионизирующих электромагнитных излучений

| Наименование параметра   | Возможные значения |
|--|--------------------|
| На расстоянии 50см от грани видеомонитора  | 10В/м              |
| На расстоянии 50см от грани видеомонитора  | 0,3А/м             |
| Для пользователей и детей дошкольных учреждений и учащихся в средних специальных и высших учебных заведениях | 20кВ/м<br>15кВ/м   |

Для уменьшения влияния этих видов излучения рекомендуется использовать мониторы с ограниченным излучением (MPR-II, TCO-92, TCO-99), установка щитов, а также сохранение регламентированных режимах труда и отдыха.

Проектирование рабочих мест, оснащенных видео-терминалами, является одним из важнейших вопросов эргономичный дизайн в области компьютерных наук.

Психологическое, физическое и антропометрические требования обязаны соответствовать на рабочем месте, и относительное положение всех ее элементов тоже. Еще один важный фактор – это характер труда. В основе, организация рабочего места программиста в начале учитывается оптимальное размещение оборудования, которое является частью рабочего места и просторное рабочее место, позволяющий выполнить все необходимые действия в кабинете.

Эргономические аспекты проектирования видео-терминалов рабочего места, а именно составление высоты рабочей грани, для пространства размещение стопы рабочего. Есть также требования для документов на рабочем столе (это наличие и размер опоры ладони, умение организовывать различные документы, расстояние от глаз пользователя до монитора, бумаг и клавиш), атрибут стола и стула на грани, контрольность элементов на рабочем столе.

Стол и кресло являются главными элементами рабочего места программиста. Основное положением это положение сидя.

Минимальную усталость вызывает у работника во время рабочих заседании положение сидя. Рациональная планировка рабочего места обеспечивает четкий порядок и последовательность размещения объектов, ручные инструменты и документацию. То, что требуется для выполнения работы, чаще всего находятся в зоне легкой досягаемости рабочего пространства.

Двигатель поле - пространство на рабочем столе, при котором правиться двигательное действие работника.

Максимальное расположение руки - это часть рабочего места, ограниченного дугами описано максимальное вытянутое руками движение плечевого размаха.



Оптимальная зона - часть моторного поля в рабочем столе, ограниченная дугами, описанные относительно фиксированного движения предплечья в локтевых суставах с опорой на локоть и плечо.

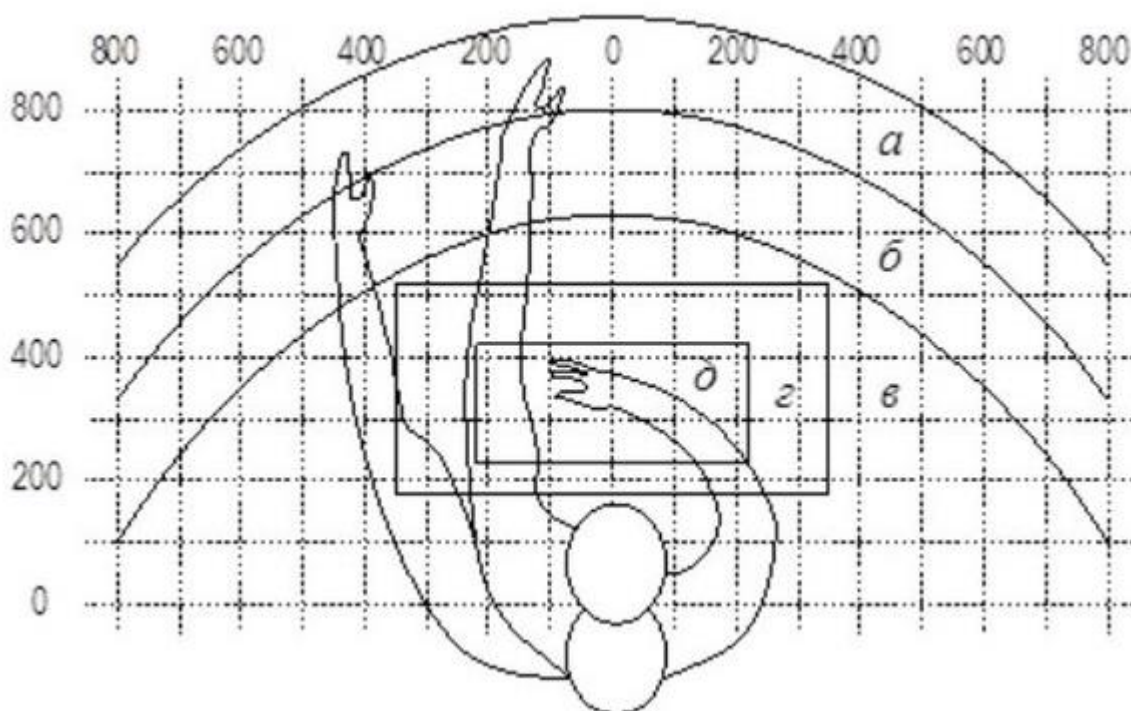


Рисунок 5.1 - Зона досягаемости рук в горизонтальной поверхности

Оптимальное размещение предметов труда и документации в зонах досягаемости:

- максимальной достигаемости изображена на зоне «а»;
- достигаемости пальцев при вытянутой руке показана в зоне «б»;
- легкой достигаемости ладони изображена в зоне «в»;
- оптимальное пространство для грубой ручной зона «г»;
- оптимальное пространство для тонкой ручной работы зона «д».

В зоне «а» находится дисплей (в центре), системный блок помещается в указанную мест по таблице ниже стола, клавиатура ставится в зоне «г» и «д». А мышь ставится в зоне справа, сканер в области «а» и «б» (слева) и принтер размещается в зоне (справа). Документация на работе должна быть легко расположена так, что работник мог достигать рукой до зоны «в» и в ящиках стола - литература, неиспользованного времени. На рис. 5.2 можно это увидеть, как правильно размещены основные компоненты и периферийные устройства программиста на рабочем столе персонального компьютера.

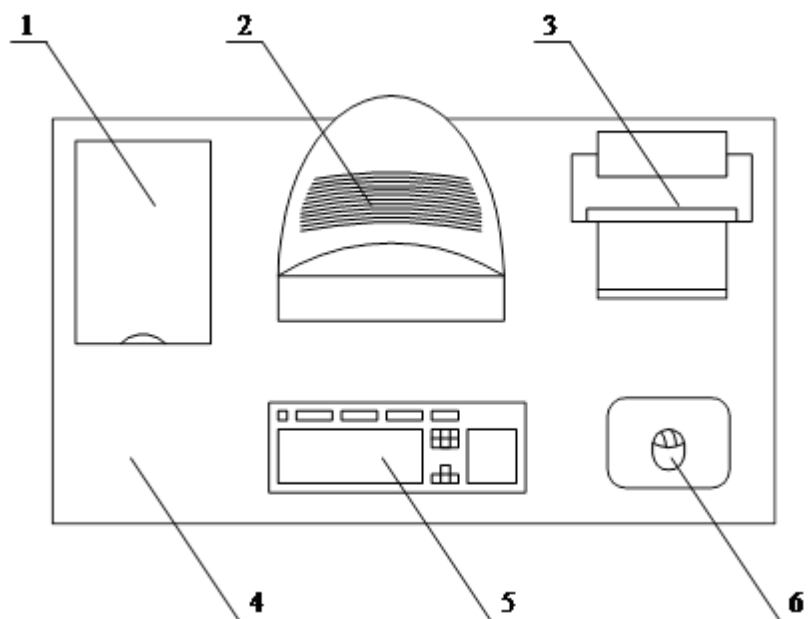


Рисунок 5.2 - Расположение основных и периферийных составляющих персонального компьютера

Расположение основных и периферийных составляющих персонального компьютера на рабочем столе программиста:

- а) сканер;
- б) монитор;
- в) принтер;
- г) грань рабочего стола;
- д) клавиатура;
- ж) мышь.

Для удобной работы программиста поверхность стола должен удовлетворять следующим требованиям:

а) высота стола должна представляться таким образом, чтобы учитывая возможность, сидеть свободно в удобном расположении, не было необходимостью основываться на подлокотники;

б) в нижней части раздела должны быть рассчитаны места для того, чтобы программисту удобно было расположиться, и не было нужды поджимать ноги;

в) в поле зрения работающего грань, на которой он работает должна обладать таким свойствам, чтобы было возможным предотвратить появление бликов;

г) стол должен быть сконструирован так, чтобы включать в себя несколько выдвижных ящиков (не менее 3-х, где можно будит хранить документы, списки и канцтовары);

д) рекомендованы пределы для высоты рабочей грани от 680 до 760 мм. Высота грани, в которой располагается клавиатура, которая не должна превышать - 650 мм.

Большое внимание уделяется для характеристики стола и стула работающего. Таким образом, оптимальная высота сиденья над полом должна располагаться в пределах 420-550 мм. Сиденья должны быть мягкими с закругленными передними краями и угол наклона спинки должен регулироваться.

Необходимо обеспечить при организации расположение различных документы на поверхности стола со стороны видео терминала, а также со стороны между монитора и клавиатуры. Кроме того, в тех случаях, когда устройство отображения видео плохого качества изображения, а именно видны мелькания. Расстояние от глаз до монитора размера должно быть около 700 мм, когда расстояние от глаз до документа составляет 300-450 мм. Расстояние пользователя до экрана от глаз, как правило при высоких качествах изображения, клавиатура и документы могут быть равны.

Размещение экрана определяется:

- расстояние подсчитывается от 0,6 до 0,7 м;
- угол чтения просмотра зрения направления в 20 градусов ниже горизонтальной и на экране перпендикулярно к этому направлению;

Предусмотрена возможность управление экраном:

- 3 см в высоту;
- по склону от -10 до +20 градусов относительно вертикали;
- по левой и правой стороне.

Огромное внимание оказывает правильная рабочая поза пользователя. Когда неудобно сидеть на работе, то могут появиться боли в мышцах, суставах и сухожилиях. Условия к рабочей позе видео терминала пользователя должна удовлетворять следующие требования:

- при наклоне головы более чем на 20 градусов;
- расслабленные плечи;
- локти расположены под углом 80 - 100 градусов;
- предплечья и руки расположены в горизонтальном положении.

Причина из-за неправильной осанки пользователя обусловлены следующими факторами: отсутствие хорошей подставки для документов, высокое расположение клавиатуры, документы расставлены ниже, для того чтобы было куда расположить руки и ноги.

Для того чтобы преодолеть эти недостатки предоставляют общие рекомендации: использование мобильной клавиатуры; должны предоставить специальные инструменты для регулировки высоты стола, клавиатуры, монитора и подставки для рук.

Размеры символов являются необходимым для продуктивной и качественной работы на компьютере, расположение их плотностей, контрастность и яркость отношение символов и фона монитора. Если расстояние от глаз до дисплея оператора составляет 60 - 80 см, высота надписи должна быть не менее 3 мм, оптимальным решением в отношении ширина и высота знака с указанием 3:4, а расстояние между метками – от 15

до 20% их высоты. Соотношение яркости фона экрана и знаков характеризуется от 1:2 до 1:15.

Во время компьютерных работ медики рекомендуют установить монитор на расстоянии 50 - 60 см от глаз. Эксперты считают, что в верхней части видео должна быть на уровне глаз или ниже. Когда человек смотрит прямо, то его глаза открыты намного шире, чем когда он смотрит вниз. В этой области просмотра значительно возрастает, что приводит к обезвоживанию глаз. Кроме того, если установлен экран высокого льна и программист смотрит с широко открытыми глазами, тем самым ухудшается функция моргания. Следствие чего ваши глаза не закрыты полностью, глаза не получают достаточное количество влаги, тем самым приводит к их усталости.

Оказание благоприятных условий труда, правильный эстетический внешний вид рабочие места в промышленности имеет огромное значение, поскольку это облегчает производительность работы. А также это повышает ее инвестиционную привлекательность, что также положительно сказывается на производительности.

Неоднократно отмечалось, что при работе с вычислительной машиной, играет очень важную роль надлежащее соблюдение работы и отдыха. В противном случае у сотрудников отмечали жалобы: значительную напряженность в зрительном аппарате, боли в голове, расстройство и нарушение сна, боли шей, рук и в позвоночной области, что вызывало недовольство работой

Информация о регламентированных перерывах представлены в таблице 5.5 необходимые при работе на компьютере, в зависимости от продолжительности рабочей смены, типов и категорий работы с ВДТ (видеотерминала) и ПК (в соответствии с СанПиН 2.2.2 542-96 "Гигиенические требования к видео-дисплейному терминалу, персональным электронно-вычислительным машинами и работ").

Таблица 5.5 – Время регламентированных перерывов при работе на компьютере

| Категория работы с ВДТ или ПК | Уровень нагрузки за рабочую смену при видах работы с ВДТ |                             |                 | Общее время регламентированных перерывов, мин |                      |
|-------------------------------|--|-----------------------------|-----------------|---|----------------------|
|                               | Группа А, количество знаков                              | Группа Б, количество знаков | Группа В, часов | При 8-часовой смене                           | При 12-часовой смене |
| I                             | до 20000   | до 15000                    | до 2,0          | 30  | 70                   |
| II                            | до 40000   | до 30000                    | до 4,0          | 50  | 90                   |
| III                           | до 60000   | до 40000                    | до 6,0          | 70  | 120                  |

Перерывы введены в соответствии с конкретными медико-санитарными правилами и нормами. Время перерывов увеличивается на 30% при несоответствии фактических условий труда санитарным правилам и нормам. Все виды работ, относящиеся к работам с использованием вычислительной машиной, разделены на три группы в соответствии с СанПиН 2.2.2 546-96. Группа «А» включает в себя работу на чтение информации с экрана ВДТ или ПК с преждевременным запросом. Ввод информации включает в себя группа «Б». Творческая работа режима диалога с компьютером относится к группе «В».

Удобная мягкая мебель, расположение аквариумов, живых цветов, либо специально выделенного помещения с газоном хорошо скажутся при работе во время перерывов. Также можно заняться легкой гимнастикой, что эффективно отразится на работе программиста.

## 5.6 Расчет освещенности и уровня шума

Расчет освещения на рабочем месте сводится к выбору осветительной системы, определение необходимого количества ламп, типа и размещения. Вследствие чего, рассчитываем параметры искусственного освещения.

Лампы накаливания и люминесцентные лампы используются как искусственное освещение, которое выполняется посредством электрических источников света [1]. По сравнению с лампами накаливания люминесцентные лампы имеют существенные преимущества:

- спектральный состав света - естественный свет, очень близок к дневному свету;
- высокое КПД, характеризующее в 1,5-2 раза больше, чем по сравнению с ламп накаливания;
- светоотдача повышена в 3-4 раза, чем у ламп накаливания;
- долговечный.

Расчет по проведению освещения в комнате проводится с площадью  $15\text{ м}^2$ , шириной 5м, и высотой - 3м. Воспользовавшись методом светового потока, рассчитаем освещенность в помещении.

Чтобы определить количество светильников понадобится определить световой поток, падающий на грань по формуле:

$$F=(E*K*S*Z)/n, \quad (5.1)$$

где F - рассчитываемый световой поток, Лм;

E - минимальная освещенность в соответствии по норме, рассчитываемая в лк (по таблице). Работа программиста, в соответствии с таблицей, относится к разряду правильных работ, что следует вывод о минимальной освещенности, которая будет равна:  $E = 300\text{ Лк}$ ;

S - площадь расчета освещаемой комнаты (в моем случае  $S = 15\text{ м}^2$ );

Z - отношение средней освещенности на минимальную от 1,1 до 1,2, где  $Z = 1,1$ ;

K - коэффициент запаса, учитывающий уменьшение светового потока лампы в результате загрязнения светильников в процессе эксплуатации (значение зависит от типа помещения и проводимых в нем работ, где  $K = 1,5$ ;

n – используем фактор, выраженный как отношение светового потока, падающего на расчетные поверхности, чтобы суммарный поток всех ламп и вычисляется в долях единицы. В основном, зависящий от характеристики лампы, размера комнаты, окраски стены и потолка, соответствующий коэффициентам отражения от стены ( $P_C$ ) и потолка ( $P_{П}$ ), где значение коэффициентов равно:  $P_C=40\%$  ,  $P_{П}=60\%$ . Данные n находятся по таблице коэффициентов пользования различных ламп.

Вычислим индекс комнаты по формуле:

$$I=S/(h*(A+B)), \quad (5.2)$$

где S - площадь комнаты,  $S = 15 \text{ м}^2$ ;

h - высота подвеса,  $h = 2.92 \text{ м}$ ;

A - ширина комнаты,  $A = 3 \text{ м}$ ;

B - длина комнаты,  $B = 5 \text{ м}$ .

Подставляя данные получим:

$$I=15/(2,92*(3+5))= 0,64$$

Зная индекс комнаты I, по таблице 5.5 определим  $n = 0,22$ .

Подставляя все данные в формулу определим поток света F:

$$F=(300*1,5*15*1,1)/0,22= 33750 \text{ Лм.}$$

Для освещения помещения возьмем люминесцентные лампы типа ЛБ40-1, световой поток которого  $F = 4320 \text{ Лк}$ . Проведем расчет необходимый для определения количество ламп по формуле:

$$N= F/F_{л}, \quad (5.3)$$

где N - число ламп;

F - световой поток,  $F = 33750 \text{ Лм}$ ;

$F_{л}$ - световой поток лампы,  $F_{л} = 4320 \text{ Лм}$

$$N= 33750/4320 = 8 \text{ штук.}$$

При постановке осветительных ламп нужно использовать светильники типа ОД. Где в комплекте каждый имеет две лампы.

Одним из неблагоприятных факторов производственной среды в ИВЦ является высокий уровень шума, создаваемый печатными устройствами, оборудованием для кондиционирования воздуха, вентиляторами систем охлаждения в самих ПК.

Для решения вопросов о необходимости и целесообразности снижения шума необходимо знать уровни шума на рабочем месте оператора.

Уровень шума, возникающий от нескольких некогерентных источников, работающих одновременно, подсчитывается на основании принципа энергетического суммирования излучений отдельных источников [12]:

$$L_{\Sigma} = 10 \lg \sum_{i=1}^{i=n} 10^{0,1L_i}, \quad (5.4)$$

где  $L_i$  – степень  $i$ -го источника шума звукового давления;

$n$  – количество источников шума.

Полученные результаты расчета сравниваются с допустимым значением уровня шума для данного рабочего места. Если результаты расчета выше допустимого значения уровня шума, то необходимы специальные меры по снижению шума. К ним относятся: облицовка стен и потолка зала звукопоглощающими материалами, снижение шума в источнике, правильная планировка оборудования и рациональная организация рабочего места оператора.

Уровни звукового давления источников шума, действующих на оператора на его рабочем месте представлены в таблице 5.6.

Таблица 5.6 - Уровни звукового давления различных источников

| Источник шума | Уровень шума, дБ |
|---------------|------------------|
| Жесткий диск  | 40               |
| Вентилятор    | 45               |
| Монитор       | 17               |
| Клавиатура    | 10               |
| Принтер       | 45               |
| Сканер        | 42               |

Обычно рабочее место оператора оснащено следующим оборудованием: винчестер в системном блоке, вентиляторы систем охлаждения ПК, монитор, клавиатура, принтер и сканер.

Подставив значения уровня звукового давления для каждого вида оборудования в формулу, получим:

$$L_{\Sigma} = 10 \cdot \lg(10^4 + 10^{4,5} + 10^{1,7} + 10^1 + 10^{4,5} + 10^{4,2}) = 49,5 \text{ дБ.}$$

Результирующее значение для рабочего места не превышает уровень шума инженера 65 дБ (ГОСТ 12.1.003-83). А если учесть еще, что периферийные устройства, такие как сканер и принтер не будут использоваться одновременно, эта цифра еще снизится. Также при работе с принтером не обязательно непосредственного присутствия оператора, так как принтер устроен с подачей листов.

В этом разделе дипломного проекта были расписаны оптимальные условия к рабочему месту инженера – программиста. Предоставленные условия труда обеспечивают удобную работу для работника. Основываясь на изученную литературу по данному вопросу, были указаны ряд оптимальных условия труда на рабочем месте программиста были рассчитаны размеры рабочей поверхности, а также провели выбор подходящей системы. Провели расчеты по оптимальному освещению производственного кабинета и проведен расчет степень влияние шума на рабочем месте. Соответствие с этими условиями, которые определяют оптимальную организацию рабочего места инженера - программиста для сохранения хорошей работоспособности в течение рабочего дня, что способствует росту, как в количественном, так и в качественном выражении производительности труда программиста, а это в первую очередь благоприятно скажется в разработки и отладки программного обеспечения.



## Заключение

Это веб-система была направлена, чтобы помочь членам программного комитета справиться со сложностью процесса судейства и проведения конференции. Система позволяет производить автоматическую генерацию сборника в нужном специальном формате.

Также система обрабатывает и собирает научные работы и исследования преподавателей, тем самым помогая в выборе консультанта и руководителя для различных работы по проекту и дипломных работ.

Система состоит из нескольких функций:

- управление конференциями позволяет создавать и управлять конференцией и семинарами. Администратор расписывает важные даты. Участники регистрируются на конференцию и отправляют свои статьи;

- управление онлайн конкурсами легко проводить разные онлайн конкурсы по проектам;

- модуль учителя - это база и профиль, где загружаются научные статьи и труды. Это поможет студентам найти подходящего руководителя.

Также продумала работу на будущее для улучшения веб-системы и добавления новых функционалов и модулей. Сначала добавление рейтинга рецензентов. Во-вторых, хочу добавить нечетную логику в “Умный Поиск”. В-третьих, было бы лучше, если система была интегрирована с системами онлайн денежных переводов (PayPal). Так как некоторые конференции не бесплатные, и участники могли бы оплатить взнос на участие через интернет.

## Список использованной литературы

1. Wikipedia – The Free Encyclopedia. URL:<https://en.wikipedia.org/wiki/>
2. Маклафлин, Б., Pollice, Г., Запад, Д. Head First объектно-ориентированного анализа и проектирования. - O'Reilly Media, Inc - 2006.
3. Официальный сайт MySQL. Официальный сайт MySQL. URL: <http://dev.mysql.com/doc/refman/5.1/en/what-is-mysql.html>
4. Петр Чен, Питер Пин-Шань. Сущность-связь Модель -. Пути к унифицированного представления данных "АСМ Сделки на системы баз данных -. 1976.
5. Рогачев. Обобщенный Model-View Controller, 2007.
6. Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влссидес. Приёмы объектно-ориентированного проектирования. Паттерны проектирования. – 2001.
7. Stripes framework documentation. URL: <http://stripes.sourceforge.net/docs>
8. Грег Хинкл. Полосы Главная. URL: <http://www.stripesframework.org/display/stripes/Home>
10. Крокфорд Дугласа. Язык Java программирования стал самый популярный в мире. - 2008.
11. iText Home page. URL: <http://itextpdf.com/>
12. Веб-разработчики Примечания. Что такое веб-сервер? URL: [http://www.webdevelopersnotes.com/basics/what\\_is\\_web\\_server.php](http://www.webdevelopersnotes.com/basics/what_is_web_server.php)
13. Расчет уровня шума. URL: <http://www.electrosad.ru/Ohlajd/noise.htm>
14. Eich, Brendan Popularity (англ.). - 3 Апрель 2008.
15. Выходное Чарльз. "Java Script: Проектирование Язык за 10 дней". Компьютер (Компьютерное общество IEEE). - 2012.
16. Белоног В.С., Калиева А.Б. Санитарные правила и нормы санпин. Алматы, 1996
17. Болодулин А.Н., Кузнецов В.Н., Мельник М.В. Теория экономического анализа: Учебное пособие. 1-е издание. Тверь: ТГТУ, 2005 г.
18. Юдин Е.Я., Борисов Л.А. Борьба с шумом на производстве: Справочник.- М.: Машиностроение, 1985 г. – 400 с.
19. Скотт Хокинс. Администрирование Web-сервера Apache и руководство по электронной коммерции. Вильямс. 2001. – 330 с.
20. Предлагаемый ECMAScript 4-е издание - Язык Обзор. - Adobe Systems, Inc-2007.
21. JQuery Статистика использования. URL: <http://trends.builtwith.com/javascript/JQuery>

## Приложение А

### *Листинг программы*

<%--

Created by IntelliJ IDEA.

Time: 3:19 AM

To change this template use File | Settings | File Templates.

```

--%>
<% @ page contentType="text/html" language="java" pageEncoding="utf-8"%>
<% @ include file="/WEB-INF/jsp/layout/include_taglibs.jspf" %>
<stripes:layout-render          name="/WEB-INF/jsp/layout/standard-main.jsp"
title="Конференция">
<stripes:useActionBean beanclass="actions.ConferenceActionBean" var="cbean"/>
<stripes:useActionBean          beanclass="actions.ManageConferenceActionBean"
var="mbean"/>
<link href="{ctx}/style/style.css" rel="stylesheet" type="text/css" />
<script type="text/javascript" src="{ctx}/js/validation.iin.js"></script>
<script src="{ctx}/js/chosen.jquery.min.js" type="text/javascript"></script>
<script                          src="{ctx}/js/scriptaculous.js?load=effects"
type="text/javascript"></script>
<stripes:layout-component name="header">
</stripes:layout-component>
<stripes:layout-component name="contents">
  <script type="text/javascript">
    jQuery(document).ready(function() {

      });

  </script>
  <style type="text/css">
    .clicked {
      color:floralwhite;
    }
  </style>
  <!-- start header -->
  <div id="header">
    <c:choose>
      <c:when test="{cbean.currentUser eq null}">
        <p><stripes:link  beanclass="actions.LoginActionBean"  style="text-
decoration:none;">Войти</stripes:link></p>
      </c:when>
      <c:otherwise>
        <p>
          ${fn:escapeXml(cbean.currentUser.firstName)}
          ${fn:escapeXml(cbean.currentUser.lastName)}
          <stripes:link  beanclass="actions.LoginActionBean"  event="logout"
style="text-decoration:none;margin-left:15px">Выход</stripes:link>
        </p>
      </c:otherwise>
    </c:choose>
  </div>

```

```

<!-- end header -->
<!-- star menu -->
<div id="menu" style="margin-left: 400px;">
  <ul class="current_page_item">
    <li><stripes:link beanclass="actions.DispatchActionBean" style="text-decoration:none;">Главная</stripes:link></li>
    <li><stripes:link beanclass="actions.ConferenceActionBean" style="text-decoration:none;">Конференции</stripes:link></li>
    <li><stripes:link beanclass="actions.CompetitionActionBean" style="text-decoration:none;">Конкурсы</stripes:link></li>
    <li><stripes:link beanclass="actions.TeacherActionBean" style="text-decoration:none;" class="arhiveList">Преподаватели</stripes:link></li>
    <li><stripes:link beanclass="actions.ShowContactsActionBean" style="text-decoration:none;">Контакты</stripes:link></li>
  </ul>
</div>
<!-- end menu -->
<!-- start page -->
<div id="page">
  <!-- start content -->
  <div id="content">
    <div class="post">
      <div class="title">
        <!-- <h2><stripes:link beanclass="actions.ShowContactsActionBean"
event="information" style="text-decoration:none;">Информационное
письмо</stripes:link></h2> -->
      </div>
      <stripes:messages />
      <stripes:errors globalErrorsOnly="true"/>
      <div class="entry">
        <p align="center"> <strong> International Students Projects
Competition <br/> on Information Technology <br/> Almaty, May 4,
2014</strong> </p>
        <p align="center"> <strong>6-ой Международный Конкурс
Студенческих проектов <br/> по информационным технологиям <br/> 4 мая
2014 года </strong></p>
        <p>
          Факультет информационных технологий (ФИТ) АУЭС извещает
студентов (бакалавров) вузов, обучающихся по специальностям, связанным с
информационными технологиями, о Международном конкурсе проектов по
информационным технологиям, который пройдет в г.Алматы в АУЭС 4 мая
2014 года. На конкурс принимаются студенческие проекты по следующим
секциям:
        </p>

```

- Мультимедийные технологии и игры** (Multimedia Technologies and Game Development);
- Проектирование баз данных, Web-приложений** (Database Design, Web-applications);
- Встроенные системы: роботы, датчики.** (Embedded Systems);
- Приложения ИТ в отраслях** (IT- applications in the industry).

Лучшие работы студентов (бакалавров) будут отмечены ценными призами, дипломами. <br/>

В рамках конкурса планируется проведение конференции научных работ магистрантов. Участие магистров в конференции может быть заочным. Оргкомитет планирует издание сборника материалов конкурса и конференции ко дню открытия конкурса. Присланные студентами статьи должны представлять краткое изложение цели исследования, методики их проведения, анализ полученных результатов. Рабочие языки презентаций и статей - казахский, русский, английский. Количество статей от одного автора должно быть не более двух. Программы-приглашения на конкурс будут высланы по электронной почте студентам, работы которых будут приняты комиссией для участия в конкурсе.

Для участия в конкурсе, необходимо до 27 апреля 2014г:

- Пройти регистрацию на сайте [aues.kz](http://www.science.kbtu.kz)
- Выслать на email [itprojectcompetition@gmail.com](mailto:itprojectcompetition@gmail.com) статью (на казахском, русском или английском языке) от 3 до 6 страниц (A5 формат) в строго указанном формате (см. [приложение 1](#) и прикрепленные образцы).

**В теме письма** обязательно указать **название секции**, например, «Multimedia Technologies and Game Development» или «Database Design, Web-applications» и т.д. <br/>

Подробности (программы, фотографии, результаты) о прошедших ИТ-конкурсах в АУЭС Вы можете увидеть на сайте <http://olympiads.kz/it-projects> <br/>

Председатель оргкомитета **Умаров Тимур Фаридович** - декан ФИТ; члены оргкомитета - преподаватели и студенты ФИТ.

В составе жюри конкурса - преподаватели НУ, СДУ, МУИТ, КазНТУ, и др. вузов РК.

```
</p>
<div id="contactInfo" style="margin-top:-10px;">
  <div><i>Контактные телефоны: +7(727)293-08-37, Кафедра
компьютерной инженерии</i></div>
  <div style="margin-left:170px;margin-top:5px;"><i>+7(727)272-46-
65, факультет информационных технологий.</i></div>
  <div style="margin-top:10px;"><i>Адрес: 050000, г.Алматы,
ул.Байтурсынова,126, АУЭС, корпус С, 123 каб.</i></div>
</div>
```

```
<a name="attachment1">
  <p align="right">Приложение 1</p>
</a>
  <p align="center"><strong>Правила оформления
статьи:</strong><br/></p>
```

```
<p style="margin-top:-15px;">
  Статьи набираются в редакторе Microsoft Word в формате А5.
Шрифт Times New Roman, размер шрифта – 10 pt, межстрочный интервал –
одинарный. Красная строка – отступ – 0,7 мм. Поля: левое - 20 мм, правое,
верхнее и нижнее – 20 мм. Выравнивание по ширине. <br/>
```

Заголовок печатается прописными буквами полужирным шрифтом – 10 pt и выравнивается по центру, переносы в заголовке не допускаются. После заголовка – пустая строка, затем по центру И.О. и Фамилия(и) автора(ов). Через 1 интервал - название организации(й) курсивом. На следующей строке ученая степень, ученое звание И.О.Фамилия научного(ых) руководителя(ей), центрировать. Ниже через 1 интервал – аннотация (только на английском языке) из 2-4 предложений курсивом 9 pt . Через один интервал- основной текст. Формулы оформлять в Equation. Рисунки (скриншоты) - в формате JPEG, в 1 статье не более 2 рисунков. После основного текста через 1 интервал список литературы, не более 3-4 источников (см. образец).

```
</p>
  <p><strong>Образцы приведены в прикрепленном файле <a
href="http://olympiads.kz/it-projects/sites/default/files/Example_A5.doc"
style="text-decoration:none;"><strong>«Образцы статей_А5»</a>.</strong></p>
```

```
</div>
</div>
</div>
<!-- end content -->
<!-- start sidebar -->
<div id="sidebar">
  <div id="search">
    <h2>Поиск</h2>
```

```

    <stripes:form beanclass="actions.SearchActionBean" id="searchForm">
        <stripes:text name="keyword" id="keyword" size="30" />
        <input type="image" src="{ctx}/images/search.png"
onclick="submitJavaScriptForm('searchForm');" style="height:20px;"/>
    </stripes:form>
</div>
<ul style="margin-top: 10px;">
    <li id="conferences">
        <h2>
            <stripes:link beanclass="actions.ConferenceActionBean"
event="conference_page" style="text-decoration:none;">
                <stripes:param name="conferenceId"
value="{mbean.conference.id}"/>
                ${fn:escapeXml(mbean.conference.name)}
            </stripes:link>
        </h2>
        <ul>
            <li><stripes:link
beanclass="actions.ManageConferenceActionBean" style="text-decoration:none;">
                <stripes:param name="conferenceId"
value="{mbean.conference.id}"/>
                Главная
            </stripes:link>
        </li>
            <li><stripes:link beanclass="actions.UploadSubmissionActionBean"
style="text-decoration:none;">Регистрация</stripes:link></li>
            <li><stripes:link
beanclass="actions.ManageConferenceActionBean" event="results" style="text-
decoration:none;">
                <stripes:param name="conferenceId"
value="{mbean.conference.id}"/>
                Результаты
            </stripes:link>
        </li>
            <li><stripes:link beanclass="actions.ShowContactsActionBean"
style="text-decoration:none;">Контакты</stripes:link></li>
        </ul>
    </li>
</ul>
<div id="fb-root"></div>
<div class="fb-like-box" style="margin: 10px 0;" data-
href="http://www.facebook.com/online.services.kz" data-width="190" data-
height="335" data-show-faces="true" data-stream="false" data-
header="false"></div>

```

```
</div>
<!-- end sidebar -->
<div id="extra" style="clear: both;">&nbsp;</div>
</div>
<!-- end page -->
<!-- start footer -->
<div id="footer">
  <jsp:useBean id="date" class="java.util.Date" />
  <p class="legal"> &copy; <fmt:formatDate value="{date}" pattern="yyyy"
/> Алматы - 2014. Все права сохранены.
</div>
<!-- end footer -->
</stripes:layout-component>
</stripes:layout-render>
```

```
<%--
Created by IntelliJ IDEA.
Time: 12:13 AM
To change this template use File | Settings | File Templates.
--%>
<%@ page contentType="text/html" language="java" pageEncoding="utf-8"%>
<%@ include file="/WEB-INF/jsp/layout/include_taglibs.jspf" %>
```



```

<stripes:useActionBean beanclass="actions.AdminHandleSubmissionsActionBean"
var="adminBean"/>
<stripes:layout-render name="/WEB-INF/jsp/layout/standard-main.jsp"
title="Панель администратора">
<link href="{ctx}/style/style.css" rel="stylesheet" type="text/css" />
<script type="text/javascript" src="{ctx}/js/validation.iin.js"></script>
<script src="{ctx}/js/chosen.jquery.min.js" type="text/javascript"></script>
<script src="{ctx}/js/scriptaculous.js?load=effects"
type="text/javascript"></script>
<stripes:layout-component name="header">
</stripes:layout-component>
<stripes:layout-component name="contents">
<script type="text/javascript">
    jQuery(document).ready(function() {
        jQuery('.arhiveList').hover(function() {
            jQuery('#archives').show();
        }, function() {
            jQuery('#archives').hide();
        });
    });
</script>
<style type="text/css">
    .clicked {
        color:floralwhite;
    }
</style>
<!-- start header -->
<div id="header">
</div>
<!-- end header -->
<!-- star menu -->
<div id="menu" style="margin-left: 400px;">
    <ul class="current_page_item">
        <li><stripes:link beanclass="actions.DispatchActionBean" style="text-
decoration:none;">Главная</stripes:link></li>
        <li><stripes:link beanclass="actions.RegisterForKbtuActionBean"
style="text-decoration:none;">Регистрация</stripes:link></li>
        <li><stripes:link beanclass="actions.ShowResultsActionBean" style="text-
decoration:none;">Результаты</stripes:link></li>
        <li><stripes:link beanclass="actions.ShowArchiveActionBean" style="text-
decoration:none;" class="arhiveList">Архив</stripes:link></li>

```

```

        <li><stripes:link beanclass="actions.ShowContactsActionBean" style="text-
decoration:none;">Контакты</stripes:link></li>
    </ul>
    <div style="margin-left: 340px;background: url(/images/hover-list.jpg) no-
repeat;position:absolute;">
        <ul id="archives" style="display:none;margin-left: 20px;">
            <li style="display:block;"><stripes:link
beanclass="actions.ShowArchiveActionBean" style="text-
decoration:none;">2012</stripes:link></li>
            <li style="display:block;"><stripes:link
beanclass="actions.ShowArchiveActionBean" event="showArchive2011"
style="text-decoration:none;">2011</stripes:link></li>
        </ul>
    </div>
</div>
<!-- end menu -->
<!-- start page -->
    <stripes:messages />
    <stripes:errors globalErrorsOnly="true"/>
    <table cellpadding="0" cellspacing="0" style="width:1000px;margin-
left:250px;">
        <tr>
            <td style="8%;">Номер</td>
            <td style="17%;">ФИО</td>
            <td style="15%;">Название доклада</td>
            <td style="35%;">Секция</td>
            <td style="10%;">E-mail</td>
            <td style="15%;">Статья</td>
        </tr>
        <c:forEach items="{adminBean.authorsList}" var="item"
varStatus="count">
            <tr>
                <td>${item.id}</td>
                <td>${fn:escapeXml(item.authors)}</td>
                <td>${fn:escapeXml(item.reportName)}</td>
                <td>${fn:escapeXml(item.sectionName)}</td>
                <td>${fn:escapeXml(item.emailStudent)}</td>
                <td>
                    <stripes:link
beanclass="actions.AdminHandleSubmissionsActionBean" event="downloadPdf"
style="text-decoration:none;">
                        <stripes:param name="id" value="{item.id}"/>
                        <stripes:param name="fileName"
value="{item.submissionFilePdf}"/>
                    </stripes:link>
                </td>
            </tr>
        </c:forEach>
    </table>

```

```

                ${fn:escapeXml(item.submissionFilePdf)}
            </stripes:link>
        </td>
    </tr>
</c:forEach>
</table>
<stripes:form
beanclass="actions.AdminHandleSubmissionsActionBean"          method="post"
id="registerForm" class="clientform">
    <stripes:submit          value="Генерировать          сборник"
name="generateProceeding" class="normwidth" style="margin-top:15px;margin-
left:250px;"/>
</stripes:form>

<!-- end page -->
<!-- start footer -->
<div id="footer">
    <jsp:useBean id="date" class="java.util.Date" />
    <p class="legal"> &copy; <fmt:formatDate value="{date}" pattern="yyyy"
/> Алматы - 2014. Все права сохранены.
</div>
<!-- end footer -->
</stripes:layout-component>

```

```

<%--
Created by IntelliJ IDEA.
Time: 2:16 PM
To change this template use File | Settings | File Templates.
--%>
<%@ page contentType="text/html" language="java" pageEncoding="utf-8"%>

```

```

<%@ include file="/WEB-INF/jsp/layout/include_taglibs.jspf" %>
<stripes:layout-render          name="/WEB-INF/jsp/layout/standard-main.jsp"
title="Панель администратора">
<stripes:useActionBean        beanclass="actions.AdminConferenceActionBean"
var="abean"/>
<link href="{ctx}/style/style.css" rel="stylesheet" type="text/css" />
<script type="text/javascript" src="{ctx}/js/validation.iin.js"></script>
<script src="{ctx}/js/chosen.jquery.min.js" type="text/javascript"></script>
<script                        src="{ctx}/js/scriptaculous.js?load=effects"
type="text/javascript"></script>
<stripes:layout-component name="header">
</stripes:layout-component>
<stripes:layout-component name="contents">
  <script type="text/javascript">
    jQuery(document).ready(function() {

      });

</script>
<style type="text/css">
  .clicked {
    color:floralwhite;
  }
</style>
<!-- start header -->
<div id="header">
  <c:choose>
    <c:when test="{abean.currentUser eq null}">
      <p><stripes:link  beanclass="actions.LoginActionBean"  style="text-
decoration:none;">Войти</stripes:link></p>
    </c:when>
    <c:otherwise>
      <p>
        ${fn:escapeXml(abean.currentUser.firstName)}
        ${fn:escapeXml(abean.currentUser.lastName)}
        <stripes:link  beanclass="actions.LoginActionBean"  event="logout"
style="text-decoration:none;margin-left:15px">Выход</stripes:link>
      </p>
    </c:otherwise>
  </c:choose>
</div>
<!-- end header -->
<!-- star menu -->
<div id="menu" style="margin-left: 400px;">

```

```

<ul class="current_page_item">
  <li><stripes:link   beanclass="actions.DispatchActionBean"   style="text-decoration:none;">Главная</stripes:link></li>
  <li><stripes:link   beanclass="actions.ConferenceActionBean"   style="text-decoration:none;">Конференции</stripes:link></li>
  <li><stripes:link   beanclass="actions.CompetitionActionBean"   style="text-decoration:none;">Конкурсы</stripes:link></li>
  <li><stripes:link   beanclass="actions.TeacherActionBean"   style="text-decoration:none;"   class="arhiveList">Преподаватели</stripes:link></li>
  <li><stripes:link   beanclass="actions.ShowContactsActionBean"   style="text-decoration:none;">Контакты</stripes:link></li>
</ul>
<div style="margin-left: 340px;background: url(/images/hover-list.jpg) no-repeat;position:absolute;">
  <ul id="archives" style="display:none;margin-left: 20px;">
    <li style="display:block;"><stripes:link
beanclass="actions.ShowArchiveActionBean" style="text-decoration:none;">2012</stripes:link></li>
    <li style="display:block;"><stripes:link
beanclass="actions.ShowArchiveActionBean" event="showArchive2011"
style="text-decoration:none;">2011</stripes:link></li>
  </ul>
</div>
</div>
<!-- end menu -->
<!-- start page -->
<div id="page1">
  <!-- start content -->
  <div id="content" style="width:770px;background-size: 770px 10px;">
    <div class="post">
      <div class="title">
        <!-- <h2><stripes:link   beanclass="actions.ShowContactsActionBean"
event="information" style="text-decoration:none;">Информационное
письмо</stripes:link></h2> -->
      </div>
      <stripes:messages />
      <stripes:errors globalErrorsOnly="true"/>
      <div class="entry">
        <p align="center"> Информация о конференции </p>
        <p> <strong> <fn:escapeXml(abean.conference.name)>
</strong></p>
        <p>
          <strong>Название конференции: </strong>
          <fn:escapeXml(abean.conference.name)> <br/>

```

```

                <strong>Город: </strong> ${fn:escapeXml(abean.conference.city)}
<br/>
                <strong>Веб-сайт: </strong>
${fn:escapeXml(abean.conference.site)} <br/>
                <strong>E-mail: </strong>
${fn:escapeXml(abean.conference.email)} <br/>
                <strong>Страна: </strong>
${fn:escapeXml(abean.conference.country)} <br/>
                <strong>Город: </strong> ${fn:escapeXml(abean.conference.city)}
</p>
                <stripes:link          beaclass="actions.AdminConferenceActionBean"
event="showSubmissionsList" style="text-decoration:none;">
                <stripes:param          name="conferenceId"
value="${abean.conferenceId}"/>
                Список статей <br/>
                </stripes:link>
                <stripes:form          beaclass="actions.AdminConferenceActionBean"
method="post" id="registerForm" class="clientform">
                <stripes:param          name="conferenceId"
value="${abean.conferenceId}"/>
                <stripes:submit          value="Генерировать          сборник"
name="generateWithPDFCopy" class="medium blue awesome" style="margin-
top:15px;margin-left:250px;"/>
                </stripes:form>
            </div>
        </div>
    </div>
    <!-- end content -->
    <div id="extra1" style="clear: both;">&nbsp;</div>
</div>
<!-- end page -->
<!-- start footer -->
<div id="footer">
    <jsp:useBean id="date" class="java.util.Date" />
    <p class="legal"> &copy; <fmt:formatDate value="${date}" pattern="yyyy"
/> Алматы - 2014. Все права сохранены.
</div>
<!-- end footer -->
</stripes:layout-component>
</stripes:layout-render>

```

```
package actions;
```

```
import hibernate.HibernateUtil;  
import jxl.write.WriteException;  
import models.TeacherArticle;  
import models.User;
```

```

import net.sourceforge.stripes.action.*;
import org.hibernate.Session;

import javax.servlet.http.HttpServletResponse;
import java.io.*;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.List;

/**
 * Created by IntelliJ IDEA.
 * User: Elima
 * Date: 5/16/13
 * Time: 7:49 PM
 * To change this template use File | Settings | File Templates.
 */
public class TeacherActionBean extends DefaultActionBean {
    private List<User> teacherList;
    private User teacher;
    private Integer userId;
    private FileBean article;
    private List<TeacherArticle> teacherArticleList;
    private String articleName;

    @DefaultHandler
    @DontValidate
    public Resolution view(){
        return new ForwardResolution("/WEB-INF/jsp/teachers-page.jsp");
    }

    @DontValidate
    @HandlesEvent("teacher_page")
    public Resolution conference_page() {
        return new ForwardResolution("/WEB-INF/jsp/teacher.jsp");
    }

    @HandlesEvent("upload")
    public Resolution upload(){
        Session s = HibernateUtil.currentSession();
        TeacherArticle teacherArticle = new TeacherArticle();
        teacherArticle.setName(getArticle().getFileName());
        teacherArticle.setTeacherId(getUserId());
        String rootPdf =
getContext().getServletContext().getInitParameter("TEACHERS_FILES_PATH");

```



```

try
{
    article.save(new File(rootPdf + article.getFileName()));
}catch (Exception e){
    System.out.println(e);
}
try {
    s.getTransaction().begin();
    s.save(teacherArticle);
    s.getTransaction().commit();
}
catch (Exception e){
    System.out.println(e);
}
getContext().getMessages().add(new SimpleMessage("Ваша работа успешно
загружена"));
return new RedirectResolution(TeacherActionBean.class, "uploadOk");
}

@DontValidate
@HandlesEvent("uploadOk")
public Resolution uploadOk() {
    return new ForwardResolution("/WEB-INF/jsp/teacher.jsp");
}

@DontValidate
@HandlesEvent("showArticle")
public Resolution showArticle() throws Exception{
    String rootPdf =
getContext().getServletContext().getInitParameter("TEACHERS_FILES_PATH");
//String rootPdf = getContext().getServletContext().getRealPath("/");
String fileNamePdf = articleName;
InputStream is = null;
try {
    is = new FileInputStream(new File(rootPdf + articleName));
} catch (FileNotFoundException ex) {
}
return new StreamingResolution(articleName, is);
}

@DontValidate
@HandlesEvent("downloadArticle")
public Resolution downloadArticle() throws Exception {

```

```

        String rootPdf =
getContext().getServletContext().getInitParameter("TEACHERS_FILES_PATH");
//String rootPdf = getContext().getServletContext().getRealPath("/");
String contentTypeStr = "application/vnd.ms-word";
InputStream is = null;
try {
    is = new FileInputStream(new File(rootPdf + articleName));
} catch (FileNotFoundException ex) {
}
if(articleName.substring(articleName.length()-4, articleName.length()-
1).equals("pdf")){
    contentTypeStr = "application/pdf";
}
return new StreamingResolution(contentTypeStr, new
FileInputStream(rootPdf+articleName)).setFilename(articleName);
}

public List<User> getTeacherList() {
    if(teacherList == null){
        teacherList = HibernateUtil.currentSession().createQuery("from User where
userType = :teacher").setString("teacher", "Преподаватель").list();
    }
    return teacherList;
}

public void setTeacherList(List<User> teacherList) {
    this.teacherList = teacherList;
}

public User getTeacher() {
    if(teacher == null){
        if(userId != null){
            teacher = (User) HibernateUtil.currentSession().createQuery("from User
where id = :id").setInteger("id", userId).uniqueResult();
        }
    }
    return teacher;
}

public void setTeacher(User teacher) {
    this.teacher = teacher;
}

public Integer getUserId() {

```

```

    return userId;
}

public void setUserId(Integer userId) {
    this.userId = userId;
}

public FileBean getArticle() {
    return article;
}

public void setArticle(FileBean article) {
    this.article = article;
}

public List<TeacherArticle> getTeacherArticleList() {
    if(teacherArticleList == null){
        teacherArticleList = HibernateUtil.currentSession().createQuery("from
TeacherArticle where teacherId = :teacherId").setInteger("teacherId",
getUserId()).list();
    }
    return teacherArticleList;
}

public void setTeacherArticleList(List<TeacherArticle> teacherArticleList) {
    this.teacherArticleList = teacherArticleList;
}

public String getArticleName() {
    return articleName;
}

public void setArticleName(String articleName) {
    this.articleName = articleName;
}
}

```

<%--

Created by IntelliJ IDEA.

User1: Arman

Date: 1/28/13

Time: 10:27 PM

To change this template use File | Settings | File Templates.

```
--%>
<% @ page contentType="text/html" language="java" pageEncoding="utf-8"%>
<% @ include file="/WEB-INF/jsp/layout/include_taglibs.jspf" %>
<stripes:layout-render name="/WEB-INF/jsp/layout/standard-main.jsp"
title="Система управления конференцией">
<stripes:useActionBean beanclass="actions.ConferenceActionBean" var="cbean"/>
<stripes:useActionBean beanclass="actions.CompetitionActionBean"
var="compbean"/>
<link href="{ctx}/style/style.css" rel="stylesheet" type="text/css" />
<script type="text/javascript" src="{ctx}/js/validation.iin.js"></script>
<script src="{ctx}/js/chosen.jquery.min.js" type="text/javascript"></script>
<script src="{ctx}/js/scriptaculous.js?load=effects"
type="text/javascript"></script>
<stripes:layout-component name="header">
<script>(function(d, s, id) {
var js, fjs = d.getElementsByTagName(s)[0];
if (d.getElementById(id)) return;
js = d.createElement(s); js.id = id;
js.src = "//connect.facebook.net/ru_RU/all.js#xfbml=1";
fjs.parentNode.insertBefore(js, fjs);
})(document, 'script', 'facebook-jssdk');</script>
</stripes:layout-component>
<stripes:layout-component name="contents">
<script type="text/javascript">
function submitJavaScriptForm(formId){
var choicePanelForm = document.getElementById(formId);
choicePanelForm.submit();
}
jQuery(document).ready(function() {

});

</script>
<style type="text/css">
.clicked {
color:floralwhite;
}
</style>
<!-- start header -->
<div id="header">
<!-- <a href="#"></a> -->
<c:choose>
```

```

        <c:when test="\${cbean.currentUser eq null}">
            <p><stripes:link beanclass="actions.LoginActionBean" style="text-
decoration:none;">Войти</stripes:link></p> <!-- style="margin-top: -
150px;margin-right: -60px;" -->
        </c:when>
        <c:otherwise>
            <p>
                \${fn:escapeXml(cbean.currentUser.firstName)}
                \${fn:escapeXml(cbean.currentUser.lastName)}
                <stripes:link beanclass="actions.LoginActionBean" event="logout"
style="text-decoration:none;margin-left:15px">Выход</stripes:link>
            </p>
        </c:otherwise>
    </c:choose>
</div>
<!-- end header -->
<!-- star menu -->
<div id="menu" style="margin-left: 400px;">
    <ul class="current_page_item">
        <li><stripes:link beanclass="actions.DispatchActionBean" style="text-
decoration:none;">Главная</stripes:link></li>
        <li><stripes:link beanclass="actions.ConferenceActionBean" style="text-
decoration:none;">Конференции</stripes:link></li>
        <li><stripes:link beanclass="actions.CompetitionActionBean" style="text-
decoration:none;">Конкурсы</stripes:link></li>
        <li><stripes:link beanclass="actions.TeacherActionBean" style="text-
decoration:none;" class="arhiveList">Преподаватели</stripes:link></li>
        <li><stripes:link beanclass="actions.ShowContactsActionBean" style="text-
decoration:none;">Контакты</stripes:link></li>
    </ul>
</div>
<!-- end menu -->
<!-- start page -->
<div id="page">
    <!-- start content -->
    <div id="content">
        <div class="post">
            <div class="title">
                <!-- <h2><stripes:link beanclass="actions.ShowContactsActionBean"
event="information" style="text-decoration:none;">Информационное
письмо</stripes:link></h2> -->
            </div>
            <stripes:messages />
            <stripes:errors globalErrorsOnly="true"/>

```

```

    <div class="entry">
        <p align="center"> <strong>Добро пожаловать в <br/> систему
управления конференциями <br/> </strong></p> <br/>
        <p> Это система управления конференциями является гибкой,
простой в использовании, и имеет множество функции, чтобы сделать его
пригодным для различных моделей конференц.<br/>
            <ul><b>Функции, которые есть в системе:</b>
                <li> Создание и проведение конференции</li>
                <li>Создание и проведение онлайн-конкурсов</li>
                <li>Автоматическая генерация сборника семинара</li>
                <li>Список последних мероприятия</li>
                <li>Модуль Учитель </li>
            </ul>
        </p>

    </div>
</div>
</div>
<!-- end content -->
<!-- start sidebar -->
<div id="sidebar">
    <div id="search">
        <h2>Поиск</h2>
        <stripes:form beanclass="actions.SearchActionBean" id="searchForm">
            <stripes:text name="keyword" id="keyword" size="30" />
            <input
                type="image"
                src="{ ctx }/images/search.png"
onclick="submitJavaScriptForm('searchForm');" style="height:20px;"/>
        </stripes:form>
    </div>
    <ul style="margin-top: 10px;">
        <li id="categories">
            <h2>Конференции</h2>
            <ul>
                <c:forEach
                    items="{ cbean.conferenceList}"
                    var="item"
varStatus="count">
                    <li>
                        <stripes:link
                            beanclass="actions.ConferenceActionBean"
event="conference_page" style="text-decoration:none;">
                            <stripes:param name="conferenceId" value="{ item.id }"/>
                            { fn:escapeXml(item.name) }
                        </stripes:link>
                    </li>
                </c:forEach>
            </ul>
        </li>
    </ul>

```

```

</li>
<li>
  <h2>Соревнования</h2>
  <ul>
    <c:forEach items="{compbean.competitionList}" var="item"
varStatus="count">
      <li>
        <stripes:link beanclass="actions.CompetitionActionBean"
event="competition_page" style="text-decoration:none;">
          <stripes:param name="competitionId" value="{item.id}"/>
          ${fn:escapeXml(item.name)}
        </stripes:link>
      </li>
    </c:forEach>
    <li>
      <stripes:link
beanclass="actions.RegisterConferenceActionBean" style="text-decoration:none;">
        Регистрация конференции
      </stripes:link>
    </li>
    <li>
      <stripes:link
beanclass="actions.RegisterUserActionBean" style="text-decoration:none;">
        Регистрация пользователя
      </stripes:link>
    </li>
  </ul>
</li>
</ul>
<div id="fb-root"></div>
<div class="fb-like-box" style="margin: 10px 0;" data-
href="http://www.facebook.com/online.services.kz" data-width="190" data-
height="335" data-show-faces="true" data-stream="false" data-
header="false"></div>
</div>
<!-- end sidebar -->
<div id="extra" style="clear: both;">&nbsp;</div>
</div>
<!-- end page -->
<!-- start footer -->
<div id="footer">
  <jsp:useBean id="date" class="java.util.Date" />
  <p class="legal"> &copy; <fmt:formatDate value="{date}" pattern="yyyy"
/> Алматы - 2014. Все права сохранены.

```

```
</div>  
<!-- end footer -->  
</stripes:layout-component>  
</stripes:layout-render>
```

```
<%--  
Created by IntelliJ IDEA.  
Time: 10:27 PM  
To change this template use File | Settings | File Templates.  
--%>  
<%@ page contentType="text/html" language="java" pageEncoding="utf-8"%>
```



```

<%@ include file="/WEB-INF/jsp/layout/include_taglibs.jspf" %>
<stripes:layout-render          name="/WEB-INF/jsp/layout/standard-main.jsp"
title="Конференции">
<stripes:useActionBean beanclass="actions.ConferenceActionBean" var="cbean"/>
<link href="{ctx}/style/style.css" rel="stylesheet" type="text/css" />
<script type="text/javascript" src="{ctx}/js/validation.iin.js"></script>
<script src="{ctx}/js/chosen.jquery.min.js" type="text/javascript"></script>
<script                          src="{ctx}/js/scriptaculous.js?load=effects"
type="text/javascript"></script>
<stripes:layout-component name="header">
</stripes:layout-component>
<stripes:layout-component name="contents">
  <script type="text/javascript">
    jQuery(document).ready(function() {

      });

</script>
<style type="text/css">
  .clicked {
    color:floralwhite;
  }
</style>
<!-- start header -->
<div id="header">
  <c:choose>
    <c:when test="{cbean.currentUser eq null}">
      <p><stripes:link  beanclass="actions.LoginActionBean"  style="text-
decoration:none;">Войти</stripes:link></p>
    </c:when>
    <c:otherwise>
      <p>
        ${fn:escapeXml(cbean.currentUser.firstName)}
        ${fn:escapeXml(cbean.currentUser.lastName)}
        <stripes:link  beanclass="actions.LoginActionBean"  event="logout"
style="text-decoration:none;margin-left:15px">Выход</stripes:link>
      </p>
    </c:otherwise>
  </c:choose>
</div>
<!-- end header -->
<!-- star menu -->
<div id="menu" style="margin-left: 400px;">
  <ul class="current_page_item">

```

```

        <li><stripes:link   beanclass="actions.DispatchActionBean"   style="text-
decoration:none;">Главная</stripes:link></li>
        <li><stripes:link   beanclass="actions.ConferenceActionBean"   style="text-
decoration:none;">Конференции</stripes:link></li>
        <li><stripes:link   beanclass="actions.CompetitionActionBean"   style="text-
decoration:none;">Конкурсы</stripes:link></li>
        <li><stripes:link   beanclass="actions.TeacherActionBean"   style="text-
decoration:none;"   class="arhiveList">Преподаватели</stripes:link></li>
        <li><stripes:link   beanclass="actions.ShowContactsActionBean"   style="text-
decoration:none;">Контакты</stripes:link></li>
    </ul>
</div>
<!-- end menu -->
<!-- start page -->
<div id="page1">
    <!-- start content -->
    <div id="content" style="width:770px;background-size: 770px 10px;">
        <div class="post">
            <div class="title">
                <!--   <h2><stripes:link   beanclass="actions.ShowContactsActionBean"
event="information"   style="text-decoration:none;">Информационное
письмо</stripes:link></h2> -->
            </div>
            <stripes:messages />
            <stripes:errors globalErrorsOnly="true"/>
            <div class="entry">
                <p   align="center">   <strong>   Ближайшие   конференции
</strong></p>
                <c:forEach   items="\${cbean.conferenceList}"   var="item"
varStatus="count">
                    <p>
                        <stripes:link   beanclass="actions.ConferenceActionBean"
event="conference_page" style="text-decoration:none;">
                            <stripes:param name="conferenceId" value="\${item.id}"/>
                            <strong> \${fn:escapeXml(item.name)} </strong> <br/>
                        </stripes:link>
                        Город: \${fn:escapeXml(item.city)} <br/>
                        Веб-сайт: \${fn:escapeXml(item.site)}
                    </p>
                </c:forEach>

            </div>
        </div>
    </div>
</div>

```

```
<!-- end content -->
<div id="extra1" style="clear: both;">&nbsp;</div>
</div>
<!-- end page -->
<!-- start footer -->
<div id="footer">
  <jsp:useBean id="date" class="java.util.Date" />
  <p class="legal"> &copy; <fmt:formatDate value="{date}" pattern="yyyy"
/> Алматы - 2014. Все права сохранены.
</div>
<!-- end footer -->
</stripes:layout-component>
</stripes:layout-render>
```

```
package actions;
```

```
import hibernate.HibernateUtil;
import jxl.write.WriteException;
import models.TeacherArticle;
import models.User;
```

```

import net.sourceforge.stripes.action.*;
import org.hibernate.Session;

import javax.servlet.http.HttpServletResponse;
import java.io.*;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.List;

/**
 * Created by IntelliJ IDEA.
 * User: Elima
 * Date: 5/16/13
 * Time: 7:49 PM
 * To change this template use File | Settings | File Templates.
 */
public class TeacherActionBean extends DefaultActionBean {
    private List<User> teacherList;
    private User teacher;
    private Integer userId;
    private FileBean article;
    private List<TeacherArticle> teacherArticleList;
    private String articleName;

    @DefaultHandler
    @DontValidate
    public Resolution view(){
        return new ForwardResolution("/WEB-INF/jsp/teachers-page.jsp");
    }

    @DontValidate
    @HandlesEvent("teacher_page")
    public Resolution conference_page() {
        return new ForwardResolution("/WEB-INF/jsp/teacher.jsp");
    }

    @HandlesEvent("upload")
    public Resolution upload(){
        Session s = HibernateUtil.currentSession();
        TeacherArticle teacherArticle = new TeacherArticle();
        teacherArticle.setName(getArticle().getFileName());
        teacherArticle.setTeacherId(getUserId());
        String rootPdf =
getContext().getServletContext().getInitParameter("TEACHERS_FILES_PATH");

```

```

try
{
    article.save(new File(rootPdf + article.getFileName()));
}catch (Exception e){
    System.out.println(e);
}
try {
    s.getTransaction().begin();
    s.save(teacherArticle);
    s.getTransaction().commit();
}
catch (Exception e){
    System.out.println(e);
}
getContext().getMessages().add(new SimpleMessage("Ваша работа успешно
загружена"));
return new RedirectResolution(TeacherActionBean.class, "uploadOk");
}

@DontValidate
@HandlesEvent("uploadOk")
public Resolution uploadOk() {
    return new ForwardResolution("/WEB-INF/jsp/teacher.jsp");
}

@DontValidate
@HandlesEvent("showArticle")
public Resolution showArticle() throws Exception{
    String rootPdf =
getContext().getServletContext().getInitParameter("TEACHERS_FILES_PATH");
//String rootPdf = getContext().getServletContext().getRealPath("/");
String fileNamePdf = articleName;
InputStream is = null;
try {
    is = new FileInputStream(new File(rootPdf + articleName));
} catch (FileNotFoundException ex) {
}
return new StreamingResolution(articleName, is);
}

@DontValidate
@HandlesEvent("downloadArticle")
public Resolution downloadArticle() throws Exception {

```

```

        String rootPdf =
getContext().getServletContext().getInitParameter("TEACHERS_FILES_PATH");
//String rootPdf = getContext().getServletContext().getRealPath("/");
String contentTypeStr = "application/vnd.ms-word";
InputStream is = null;
try {
    is = new FileInputStream(new File(rootPdf + articleName));
} catch (FileNotFoundException ex) {
}
if(articleName.substring(articleName.length()-4, articleName.length()-
1).equals("pdf")){
    contentTypeStr = "application/pdf";
}
return new StreamingResolution(contentTypeStr, new
FileInputStream(rootPdf+articleName)).setFilename(articleName);
}

public List<User> getTeacherList() {
    if(teacherList == null){
        teacherList = HibernateUtil.currentSession().createQuery("from User where
userType = :teacher").setString("teacher", "Преподаватель").list();
    }
    return teacherList;
}

public void setTeacherList(List<User> teacherList) {
    this.teacherList = teacherList;
}

public User getTeacher() {
    if(teacher == null){
        if(userId != null){
            teacher = (User) HibernateUtil.currentSession().createQuery("from User
where id = :id").setInteger("id", userId).uniqueResult();
        }
    }
    return teacher;
}

public void setTeacher(User teacher) {
    this.teacher = teacher;
}

public Integer getUserId() {

```

```

    return userId;
}

public void setUserId(Integer userId) {
    this.userId = userId;
}

public FileBean getArticle() {
    return article;
}

public void setArticle(FileBean article) {
    this.article = article;
}

public List<TeacherArticle> getTeacherArticleList() {
    if(teacherArticleList == null){
        teacherArticleList = HibernateUtil.currentSession().createQuery("from
TeacherArticle where teacherId = :teacherId").setInteger("teacherId",
getUserId()).list();
    }
    return teacherArticleList;
}

public void setTeacherArticleList(List<TeacherArticle> teacherArticleList) {
    this.teacherArticleList = teacherArticleList;
}

public String getArticleName() {
    return articleName;
}

public void setArticleName(String articleName) {
    this.articleName = articleName;
}
}

```

<%--

Created by IntelliJ IDEA.

Time: 12:13 AM

To change this template use File | Settings | File Templates.

--%>

```

<% @ page contentType="text/html" language="java" pageEncoding="utf-8"% >
<% @ include file="/WEB-INF/jsp/layout/include_taglibs.jspf" %>
<stripes:useActionBean beanclass="actions.AdminHandleSubmissionsActionBean"
var="adminBean"/>
<stripes:layout-render name="/WEB-INF/jsp/layout/standard-main.jsp"
title="Панель администратора">
<link href="{ctx}/style/style.css" rel="stylesheet" type="text/css" />
<script type="text/javascript" src="{ctx}/js/validation.iin.js"></script>
<script src="{ctx}/js/chosen.jquery.min.js" type="text/javascript"></script>
<script src="{ctx}/js/scriptaculous.js?load=effects"
type="text/javascript"></script>
<stripes:layout-component name="header">
</stripes:layout-component>
<stripes:layout-component name="contents">
<script type="text/javascript">
    jQuery(document).ready(function() {
        jQuery('.arhiveList').hover(function() {
            jQuery('#archives').show();
        }, function() {
            jQuery('#archives').hide();
        });
    });
</script>
<style type="text/css">
    .clicked {
        color:floralwhite;
    }
</style>
<!-- start header -->
<div id="header">
</div>
<!-- end header -->
<!-- star menu -->
<div id="menu" style="margin-left: 400px;">
    <ul class="current_page_item">
        <li><stripes:link beanclass="actions.DispatchActionBean" style="text-decoration:none;">Главная</stripes:link></li>
        <li><stripes:link beanclass="actions.RegisterForKbtuActionBean" style="text-decoration:none;">Регистрация</stripes:link></li>
        <li><stripes:link beanclass="actions.ShowResultsActionBean" style="text-decoration:none;">Результаты</stripes:link></li>
    </ul>
</div>

```



```

        <li><stripes:link beanclass="actions.ShowArchiveActionBean" style="text-
decoration:none;" class="arhiveList">Архив</stripes:link></li>
        <li><stripes:link beanclass="actions.ShowContactsActionBean" style="text-
decoration:none;">Контакты</stripes:link></li>
    </ul>
    <div style="margin-left: 340px;background: url(/images/hover-list.jpg) no-
repeat;position:absolute;">
        <ul id="archives" style="display:none;margin-left: 20px;">
            <li style="display:block;"><stripes:link
beanclass="actions.ShowArchiveActionBean" style="text-
decoration:none;">2012</stripes:link></li>
            <li style="display:block;"><stripes:link
beanclass="actions.ShowArchiveActionBean" event="showArchive2011"
style="text-decoration:none;">2011</stripes:link></li>
        </ul>
    </div>
</div>
<!-- end menu -->
<!-- start page -->
    <stripes:messages />
    <stripes:errors globalErrorsOnly="true"/>
    <table cellpadding="0" cellspacing="0" style="width:1000px;margin-
left:250px;">
        <tr>
            <td style="8%;">Номер</td>
            <td style="17%;">ФИО</td>
            <td style="15%;">Название доклада</td>
            <td style="35%;">Секция</td>
            <td style="10%;">E-mail</td>
            <td style="15%;">Статья</td>
        </tr>
        <c:forEach items="\${adminBean.authorsList}" var="item"
varStatus="count">
            <tr>
                <td>\${item.id}</td>
                <td>\${fn:escapeXml(item.authors)}</td>
                <td>\${fn:escapeXml(item.reportName)}</td>
                <td>\${fn:escapeXml(item.sectionName)}</td>
                <td>\${fn:escapeXml(item.emailStudent)}</td>
                <td>
                    <stripes:link
beanclass="actions.AdminHandleSubmissionsActionBean" event="downloadPdf"
style="text-decoration:none;">
                        <stripes:param name="id" value="\${item.id}"/>

```

```

        <stripes:param                                name="fileName"
value="\${item.submissionFilePdf}"/>
        <stripes:link                                >
            <td>
                <tr>
                    </tr>
                </td>
            </tr>
        </c:forEach>
    </table>
    <stripes:form
beanclass="actions.AdminHandleSubmissionsActionBean"        method="post"
id="registerForm" class="clientform">
        <stripes:submit                                value="Генерировать        сборник"
name="generateProceeding" class="normwidth" style="margin-top:15px;margin-
left:250px;"/>
    </stripes:form>

    <!-- end page -->
    <!-- start footer -->
    <div id="footer">
        <jsp:useBean id="date" class="java.util.Date" />
        <p class="legal"> &copy; <fmt:formatDate value="\${date}" pattern="yyyy"
/> Алматы - 2014. Все права сохранены.
    </div>
    <!-- end footer -->
</stripes:layout-component>
</stripes:layout-render>

```